Índice

Índice I
1 Generalidades 1
1.1 Contenido del kit de iniciación ILC 150 ETH 1
1.2 Características del sistema 2
1.2.1 Sistema operativo 2
1.2.2 Especificaciones materiales del PC WorX 2
1.2.3 Software de programación 2
1.2.4 Firmware 2
2 Descripción del material 3
2.1 Estructura del kit de iniciación 3
2.2 Cableado del kit de iniciación 4
3 Instalación de PC WorX 6
3.1 Instalación del software 6
3.1.1 Iniciar el programa de instalación de PC WorX 6
3.1.2 Instalación de Service Packs y/o Hotfix 6
3.1.3 Iniciar PC WorX 7
3.2 Activación de la licencia básica de PC WorX 7
4 Informaciones generales sobre PC WorX 8
4.1 Algunos iconos importantes de la barra de herramientas 8
4.2 Áreas de trabajo 9
4.2.1 Área de trabajo « Programación IEC » 9
4.2.2 Área de trabajo « Configuración de Bus » 10
4.2.3 Área de trabajo « Asignación de datos de proceso » 11
4.3 Compilación y descarga 12
5 EL IEC 61131-3 13
5.1 Introducción 13
5.2 Estructura de una aplicación en función de la norma IEC 61131-3 13
5.3 POU 15
5.3.1 Funciones 15
5.3.2 Bloques de funciones 16
5.3.3 Programas 17

5.4 Tipos de datos 18
5.4.1 Tipos generales de datos 18
5.4.2 Matriz (Array) 19
5.4.3 Estructura (Struct) 21
5.4.4 Cadena (String) 21
5.5 Texto Estructurado (Structured Text) 22
5.5.1 Selección de declaraciones 22
5.5.2 Declaraciones de iteración 23
6 Creación de un proyecto 24
6.1 Resumen 24
6.2 Abrir un nuevo proyecto 25
6.3 Estructura del nuevo proyecto de acuerdo con la norma IEC 61131-3 26
6.4 Adaptación de las informaciones del proyecto 27
6.5 Configuración IP del ordenador 28
6.6 Configuración IP del controlador 30
6.6.1 Generalidades 30
6.6.2 Adaptar la dirección IP mediante BootP Server 31
6.7 La configuración INTERBUS 33
6.8 Compilación y descarga 35
7 Programa de ejemplo 36
7.1 Introducción 36
7.2 Descripción de la misión 36
7.3 Elaboración 36
7.3.1 Creación de una nueva POU 37
7.3.2 Definición de las variables 38
7.3.3 Creación de un bloque de funciones « Parpadeo » 40
7.3.4 El programa principal 42
7.3.5 Pruebas del programa 44
7.3.6 Bloque de funciones para un tratamiento de la señal analógica 45
7.3.7 Asignación de las variables de proceso 48
7.3.8 Ventana de inspecciones 49
8 Webvisit 50
8.1 Introducción 50
8.2 Inicio de la visualización 51

8.3 Añadir una modificación en WebVisit 52
8.3.1 Crear un nuevo proyecto 52
8.3.2 Crear una nueva página web 53
8.3.3 Descarga del proyecto 59
9 Trucos y recursos 60
9.1 Gestión de la pantalla 60
9.2 Gestión de las variables 61
10 Ejercicios 63
10.1 Un sistema de alarma 63
10.2 Apertura de un portón 65
10.3 Control de la temperatura 67
10.4 Ejemplo en ST con los array y las estructuras 68

www.infoPLC.net

1 Generalidades

1.1 Contenido del kit de iniciación ILC 150 ETH

El kit de iniciación contiene el ILC 150 ETH, con entradas y salidas digitales y analógicas y está completamente montado.

Descripción	Tipo	N°. de mando	Número
Software			
PC WorX	AX SW SUITE DEMO CD	2985660	1
Material (módulos)			
Controlador Inline	ILC 150 ETH	2985330	1
Módulo Inline con una salida analógica	IB IL AO 1/U/SF-PAC	2861399	1
Módulo Inline con dos entradas analógicas	IB IL AI 2/SF-ME	2863944	1
Material (accesorios)			
Potenciómetro	EMG 30-SP/10K LIN	2942124	1
Cable Ethernet, 2 m	FL CAT5 FLEX CONF	2744843	1
Alimentación	STEP-PS-100-240AC/24DC/1.5	2938947	1
Módulo de conmutación (8 conmutadores)	UM 45-IB-DI:SIM8	2962997	1
Gráfico de barras con indicador LED	ONE PART AO BAR GRAPH	-	1
Borne terminal estándar, gris	CLIPFIX 35-5	3022276	4
Borne modular de conexión por resorte (multiposiciones) gris	STS 2,5-QUATRO	3031746	1
Borne modular de conexión por resorte (multiposiciones) azul	STS 2,5-QUATRO BU	3036288	1
Protección del borne modular (gris)	D-STS 2,5	3031762	2
Documentación			
Manual del usuario	UM QS EN ILC 150 ETH STARTERKIT	2910295	1
Registro del software	PACKB.SOFTWARE REG	9013743	1
CD	CD ILC 150 ETH STARTERKIT V1.00	9036973	

www.infoPLC.net

1.2 Características del sistema

1.2.1 Sistema operativo

- Windows 2000 SP3
- Windows XP SP2 (recomendado)

1.2.2 Especificaciones materiales del PC WorX

Especificaciones materiales	
Unidad central	Pentium III 800, 1 GHz (recomendado)
Memoria	128 MB (mínimo), 256 MB (recomendado)
Disco duro	500 MB libres
Lector CD-ROM	SÍ
Interfaces	1 x serie, Ethernet
Pantalla	SVGA, resolución de 800 x 600 (mínimo) 1024 x 768 (recomendado)
Accesorios	Teclado, ratón

1.2.3 Software de programación

Software	Versión
PC WorX	≥ 5.20 SP1

1.2.4 Firmware

Módulo	Versión
ILC 150 ETH	≥ 2.00

www.infoPLC.net

2 Descripción del material

2.1 Estructura del kit de iniciación

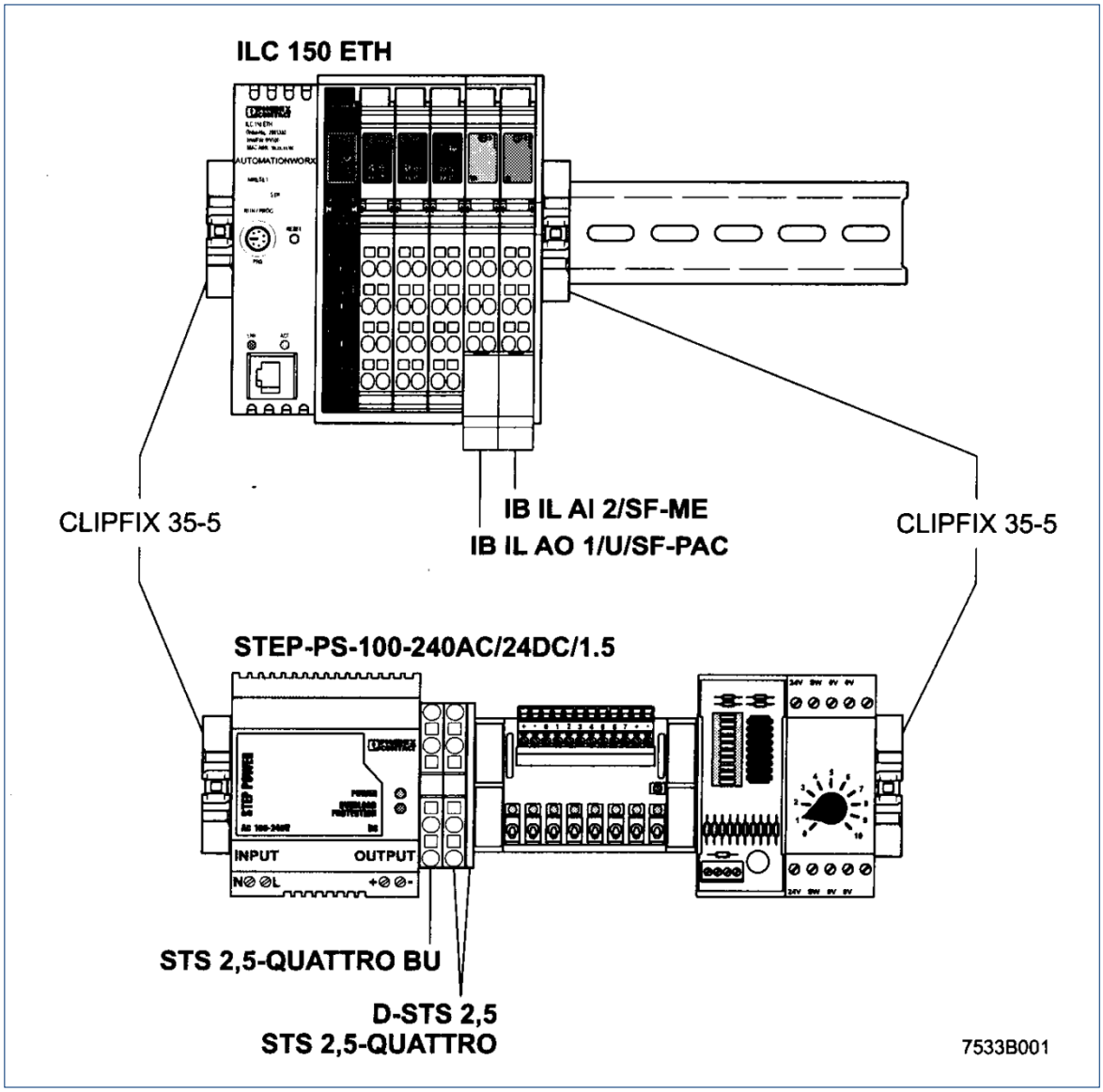


Ilustración 2-1 Estructura del kit de iniciación.

www.infoPLC.net

2.2 Cableado del kit de iniciación

Módulo	Entrada–salida	Punto de señal	Variable
ILC 150 ETH	Entrada I1	Conector 3, punto de conexión 1.1	ONBOARD_INPUT_BIT0
	Entrada I2	Conector 3, punto de conexión 2.1	ONBOARD_INPUT_BIT1
	Entrada I3	Conector 3, punto de conexión 1.4	ONBOARD_INPUT_BIT2
	Entrada I4	Conector 3, punto de conexión 2.4	ONBOARD_INPUT_BIT3
	Entrada I5	Conector 4, punto de conexión 3.1	ONBOARD_INPUT_BIT4
	Entrada I6	Conector 4, punto de conexión 4.1	ONBOARD_INPUT_BIT5
	Entrada I7	Conector 4, punto de conexión 3.4	ONBOARD_INPUT_BIT6
	Entrada I8	Conector 4, punto de conexión 4.4	ONBOARD_INPUT_BIT7
IB IL AO 1/U/SF-PAC	Salida O1	Conector 5, punto de conexión 1.1	Salida_Analogica
IB IL AI 2/SF-ME	Entrada I9	Conector 6, punto de conexión 1.1	Entrada_Analogica

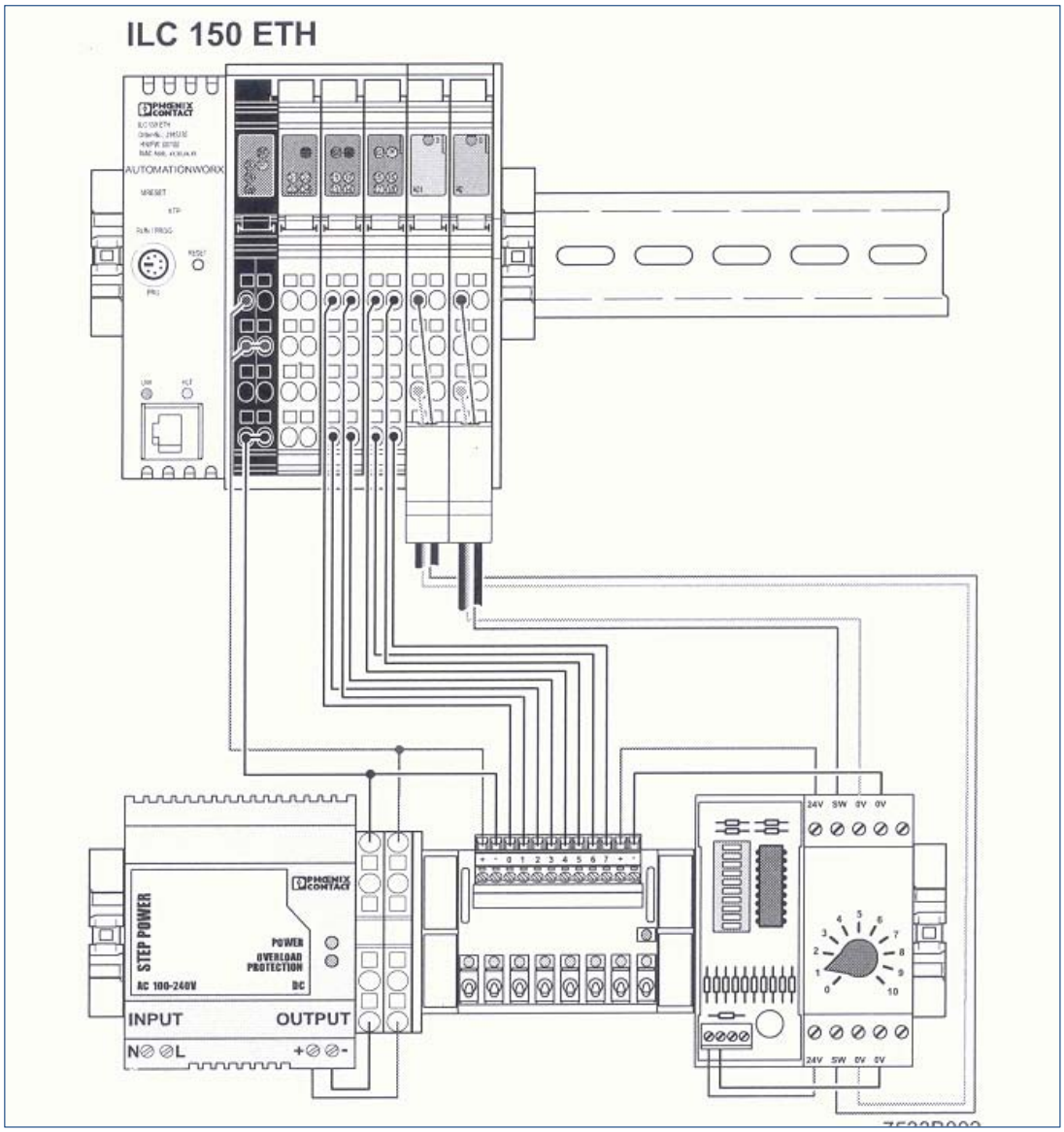


Ilustración 2-2 Cableado del kit de iniciación.

www.infoPLC.net

3 Instalación de PC WorX

3.1 Instalación del software

Para instalarlo, se deben cerrar todas las aplicaciones Windows.

Esta versión de PC WorX se puede instalar paralelamente a una versión anterior.

Primero hay que instalar el software PC WorX que se encuentra en el AX SW SUITE DEMO CD.

Instalar a continuación el Service Pack para el software PC WorX, que se encuentran en el CD ILC 150 ETH Starterkit CD.

3.1.1 Iniciar el programa de instalación de PC WorX

- Normalmente, el programa de instalación se inicia unos segundos después de haber introducido el CD-ROM en el lector CD.
  Si no fuese así, se debe iniciar mediante SETUP.EXE del directorio “[drive]:\setup\” del CD-ROM.
- Siga, a continuación, las instrucciones del programa de instalación.
- Se crean automáticamente todos los directorios necesarios y se copian todos los ficheros para hacer posible la ejecución de programas.
- Después de una instalación correcta, se ha de reiniciar el PC.

3.1.2 Instalación de Service Packs y/o Hotfix

- Tras la instalación de PC WorX, se puede realizar la instalación del Service Pack y de Hotfix que corresponda (en la página web de Phoenix Contact se encuentra siempre la versión más actual).
  Aquí también, se iniciará automáticamente el programa de instalación. Siga las instrucciones del programa de instalación.
- Tras una instalación correcta, se puede reiniciar el PC.

3.1.3 Iniciar PC WorX

- Inicie PC WorX mediante "Inicio… Todos los programas… Phoenix Contact… AUTOMATIONWORX Software suite 200x 1.xx… PC WorX 5.xx".
- Si es la primera vez que se inicia el PC WorX, éste se iniciará en modo démo, con posibilidades limitadas.
- En el apartado siguiente, abordaremos la manera en que se ha de instalar una licencia básica. Para pasar de una licencia básica a una licencia profesional, póngase en contacto con Phoenix Contact.

3.2 Activación de la licencia básica de PC WorX

Después de iniciar el PC WorX, se puede activar la licencia de la siguiente forma :

- Seleccione "Registro" en el menú "?", como aparece en la ilustración 3-1 :

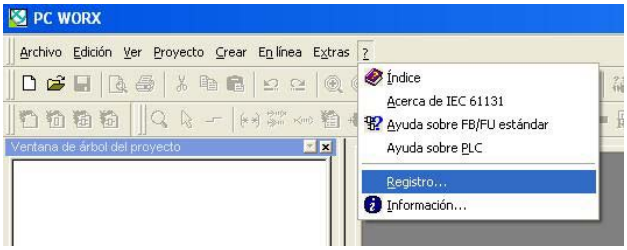


Ilustración 3-1

- Introduzca el código de registro :

> El código de registro de este kit de iniciación es :
>
> 19E80 – 19A75 – E3B30 – C895B – 8D8CF – 04541 – B5E

- Confirme con OK.

Este código de registro se activa cuando se reinicia el PC WorX.

4 Informaciones generales sobre PC WorX

4.1 Algunos iconos importantes de la barra de herramientas

De acuerdo con la normativa IEC 61131-3, se prevé una separación completa entre el hardware y el software. Esta es la razón por la cual se definen tres áreas de trabajo : un área de trabajo para establecer la configuración física (Configuración del Bus) y un área de trabajo para escribir el programa (Programación IEC). Finalmente, existe un tercer área de trabajo en que se realiza la asociación entre el software y los elementos de entrada–salida físicos.

Mediante los tres iconos siguientes, se pasa fácilmente entre los diferentes áreas de trabajo.

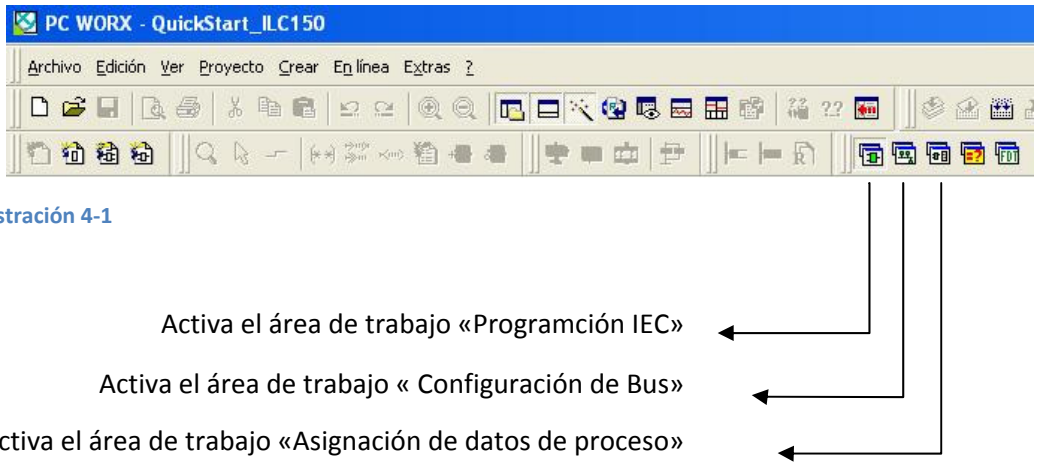

PC WORX - QuickStart_ILC150
Archivo Edición Ver Proyecto Crear En línea Extras ?
Activa el área de trabajo «Programción IEC»
Activa el área de trabajo « Configuración de Bus»
Activa el área de trabajo «Asignación de datos de proceso»

Ilustración 4-1

4.2 Áreas de trabajo

4.2.1 Área de trabajo « Programación IEC »

En el área de trabajo « Programación IEC », las siguientes ventanas son muy importantes :

- Venta de árbol de proyecto (a la izquierda) - <Shift> F8
- Ventana de programación (en medio) - <Shift> F2
- Asistente de edición (a la derecha) - <Ctrl> F2

Se pueden añadir o suprimir otras ventanas.

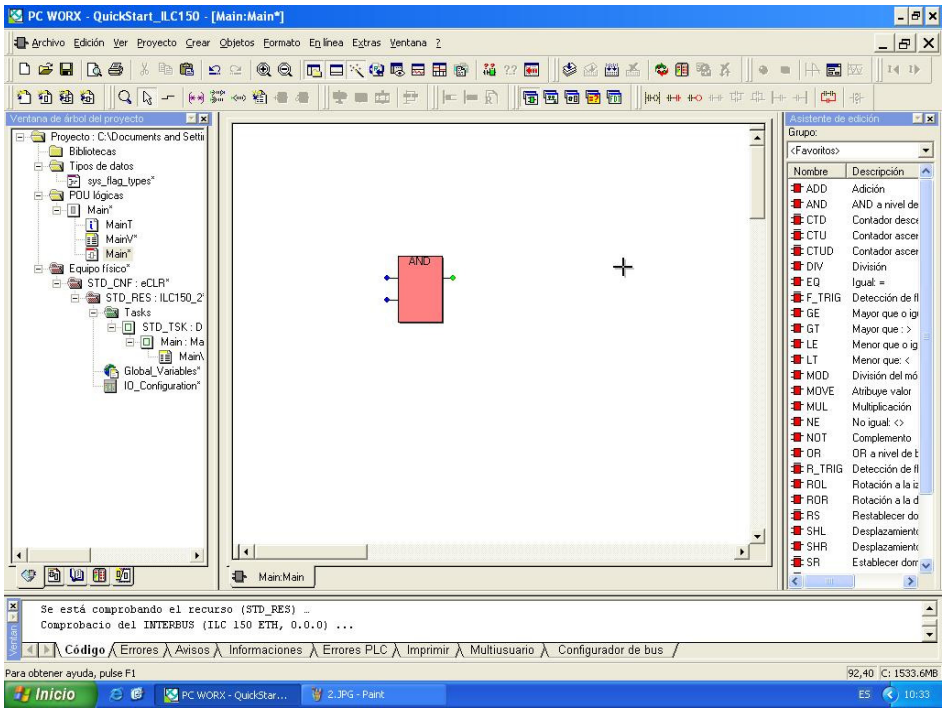

PC WORX - QuickStart_ILC150 - [Main:Main*]
Archivo Edición Ver Proyecto Crear Objetos Formato En línea Extras Ventana ?
Ventana de árbol del proyecto
Asistente de edición
Grupo:
<Favoritos>
Nombre
Descripción
AND
Main:Main
Se está comprobando el recurso (STD_RES) ...
Comprobacio del INTERBUS (ILC 150 ETH, 0.0.0) ...
Código Errores Avisos Informaciones Errores PLC Imprimir Multiusuario Configurador de bus
Para obtener ayuda, pulse F1
Inicio

Ilustración 4-2 Área de trabajo « Programación IEC ».

Si no se visualiza o se ha cerrado una ventana, siempre puede reactivarse seleccionando la ventana deseada en el elemento « Visualización ».

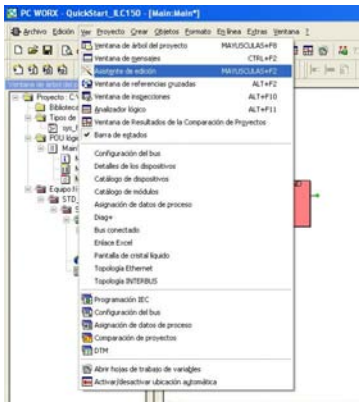


Ilustración 4-3

4.2.2 Área de trabajo « Configuración de Bus »

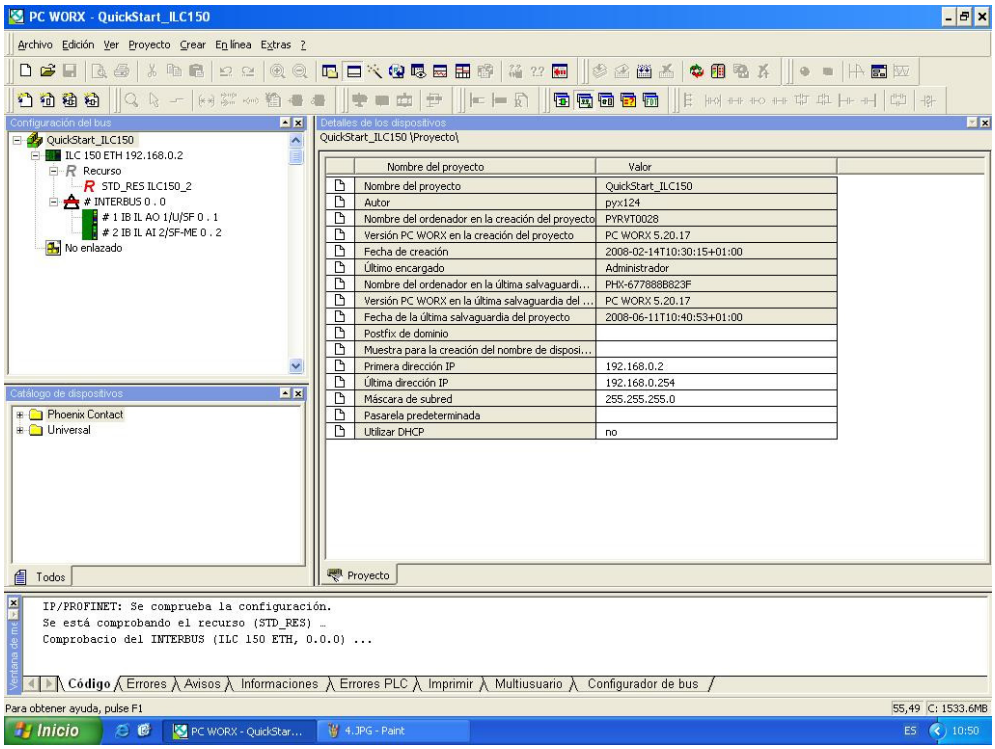

PC WORX - QuickStart_ILC150
Archivo Edición Ver Proyecto Crear En línea Extras ?
Configuración del bus
QuickStart_ILC150
ILC 150 ETH 192.168.0.2
Recurso
STD_RES ILC150_2
INTERBUS 0 . 0
1 IB IL AO 1/U/SF 0 . 1
2 IB IL AI 2/SF-ME 0 . 2
No enlazado
Detalles de los dispositivos
QuickStart_ILC150 \Proyecto\
Nombre del proyecto
Valor
Nombre del proyecto QuickStart_ILC150
Autor pyx124
Nombre del ordenador en la creación del proyecto PYRVT0028
Versión PC WORX en la creación del proyecto PC WORX 5.20.17
Fecha de creación 2008-02-14T10:30:15+01:00
Último encargado Administrador
Nombre del ordenador en la última salvaguardi... PHX-677888B823F
Versión PC WORX en la última salvaguardia del ... PC WORX 5.20.17
Fecha de la última salvaguardia del proyecto 2008-06-11T10:40:53+01:00
Postfix de dominio
Muestra para la creación del nombre de disposi...
Primera dirección IP 192.168.0.2
Última dirección IP 192.168.0.254
Máscara de subred 255.255.255.0
Pasarela predeterminada
Utilizar DHCP no
Catálogo de dispositivos
Phoenix Contact
Universal
Todos
Proyecto
IP/PROFINET: Se comprueba la configuración.
Se está comprobando el recurso (STD_RES) …
Comprobacio del INTERBUS (ILC 150 ETH, 0.0.0) ...
Código Errores Avisos Informaciones Errores PLC Imprimir Multiusuario Configurador de bus
Para obtener ayuda, pulse F1
55,49 C: 1533.6MB
Inicio
PC WORX - QuickStar...
4.JPG - Paint
ES 10:50

Ilustración 4-4 Área de trabajo « Configuración del Bus ».

En el área de trabajo « Configuración del Bus », encontramos todos los detalles referentes al hardware. En la ventana « Detalles de los dispositivos », encontramos las propiedades generales del proyecto.

En cada módulo (elemento de entradas/salidas), encontramos, en la ventana « Detalles de los dispositivos », todas las propiedades importantes, todos los ajustes y todas las fichas técnicas en pestañas diferentes, ver ilustración 4-5.

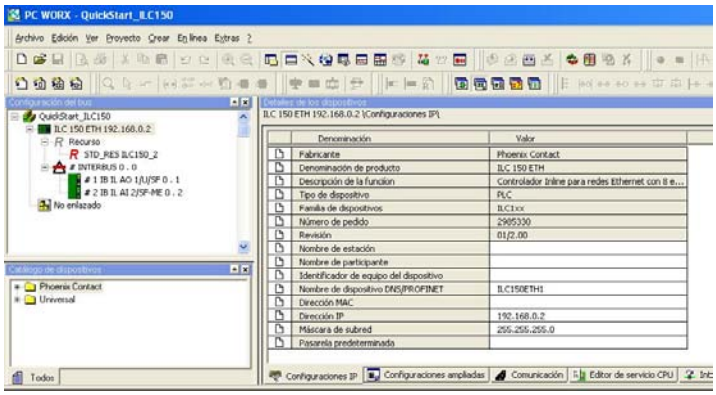


Ilustración 4-5 Detalles del dispositivo, pestañas diferentes por módulo.

4.2.3 Área de trabajo « Asignación de datos de proceso »

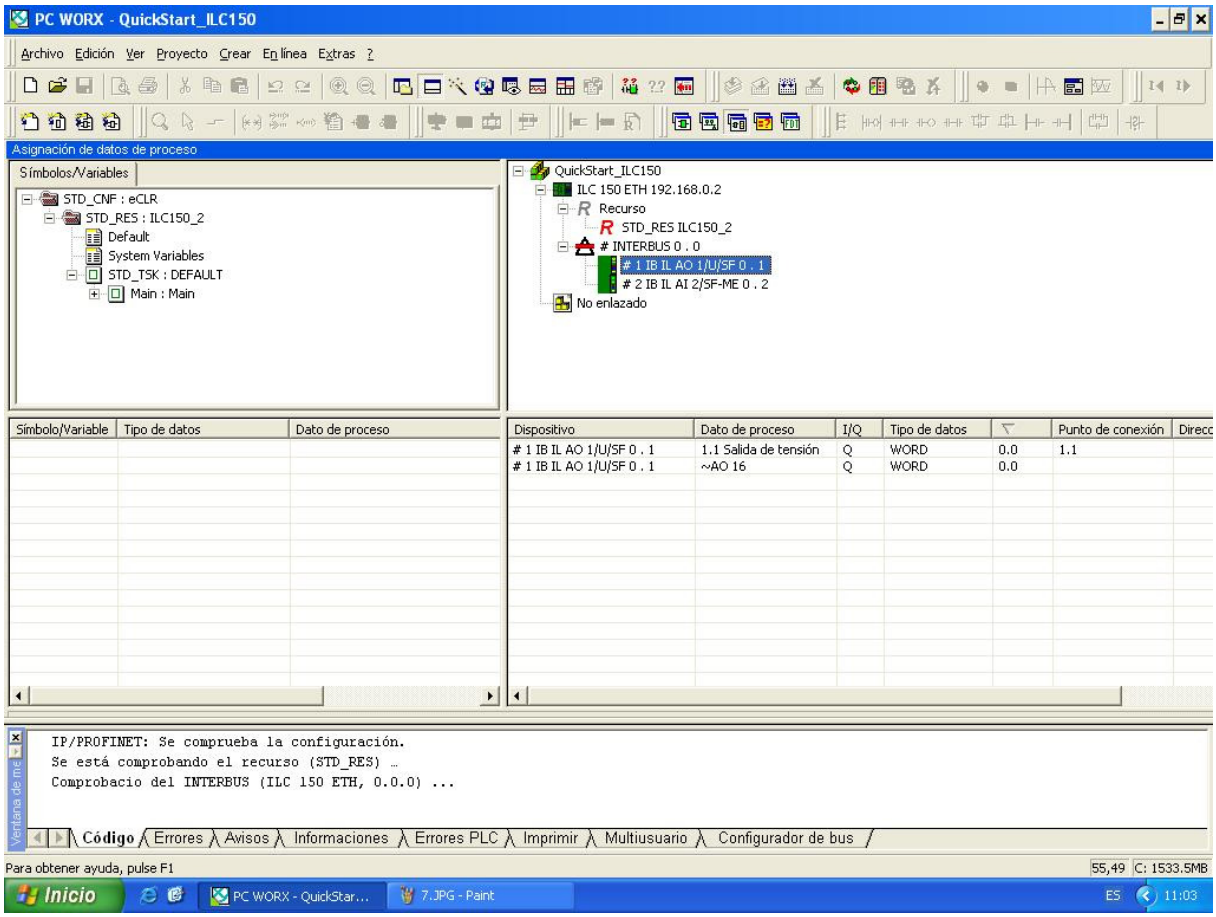


Ilustración 4-6

En el área de trabajo « Asignación de datos de proceso », existe la posibilidad de asociar las variables definidas en el software a los puntos entrada–salida disponibles en los módulos de la configuración hardware.

4.3 Compilación y descarga

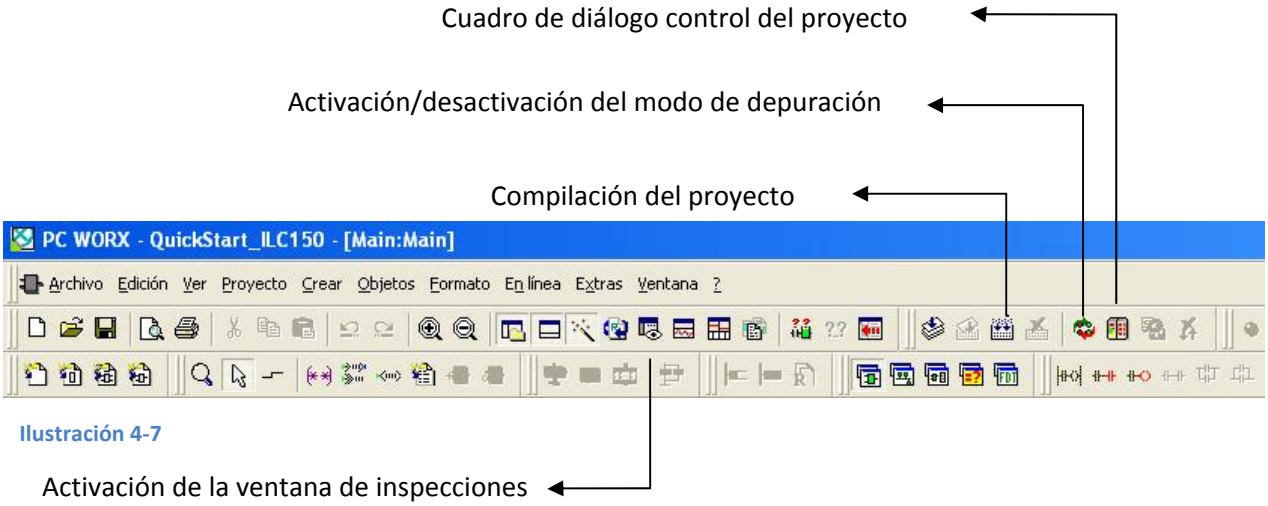

Cuadro de diálogo control del proyecto
Activación/desactivación del modo de depuración
Compilación del proyecto
PC WORX - QuickStart_ILC150 - [Main:Main]
Archivo Edición Ver Proyecto Crear Objetos Formato En línea Extras Ventana ?
Activación de la ventana de inspecciones

Ilustración 4-7

La ilustración 4-8 visualiza el “Diálogo de control de proyecto”. Mediante el mismo se puede actuar sobre los PLC. En otras palabras, los PLC se pueden detener, reiniciar, activar o desactivar, y con esta ventana, se puede también realizar la descarga de un programa.

Ilustración 4-8 Diálogo de control de proyecto.

5 EL IEC 61131-3

5.1 Introducción

La NORMA IEC 61131 es una normativa internacional para los sistemas de control programables (PLC, PLC soft, ...) y comprende 5 partes :

IEC 61131-1 : Resumen general
IEC 61131-2 : Características materiales
IEC 6113 1-3 : Lenguajes de programación
IEC 6113 1-4 : Directrices para el usuario
IEC 6113 1-5 : Comunicación
+ 2 informes técnicos.

Desde 1993, la NORMA IEC 61131-3 es la normativa internacional para la programación de los controladores industriales.
Define 5 lenguajes distintos. Estos lenguajes se utilizan en todo el mundo por los fabricantes importantes de PLC. Estos 5 lenguajes son los siguientes :

- LADDER DIAGRAM (LD), o diagrama de contactos: un lenguaje gráfico basado en la lógica del contactor.
- FUNCTION BLOCK DIAGRAM (FBD), o diagrama de bloques de funciones.
- INSTRUCTION LIST (IL), o lista de instrucciones: un lenguaje de « nivel bajo », comparable al ensamblador.
- SEQUENTIAL FUNCTION CHART (SFC), o diagrama de funciones secuenciales : un lenguaje gráfico desarrollado especialmente para programar los sistemas secuenciales.
- STRUCTURED TEXT (ST), o texto estructurado : un lenguaje de « alto nivel » para programar de forma estructurada. Esta lenguaje es muy parecido al lenguaje PASCAL.

5.2 Estructura de una aplicación en función de la norma IEC 61131-3

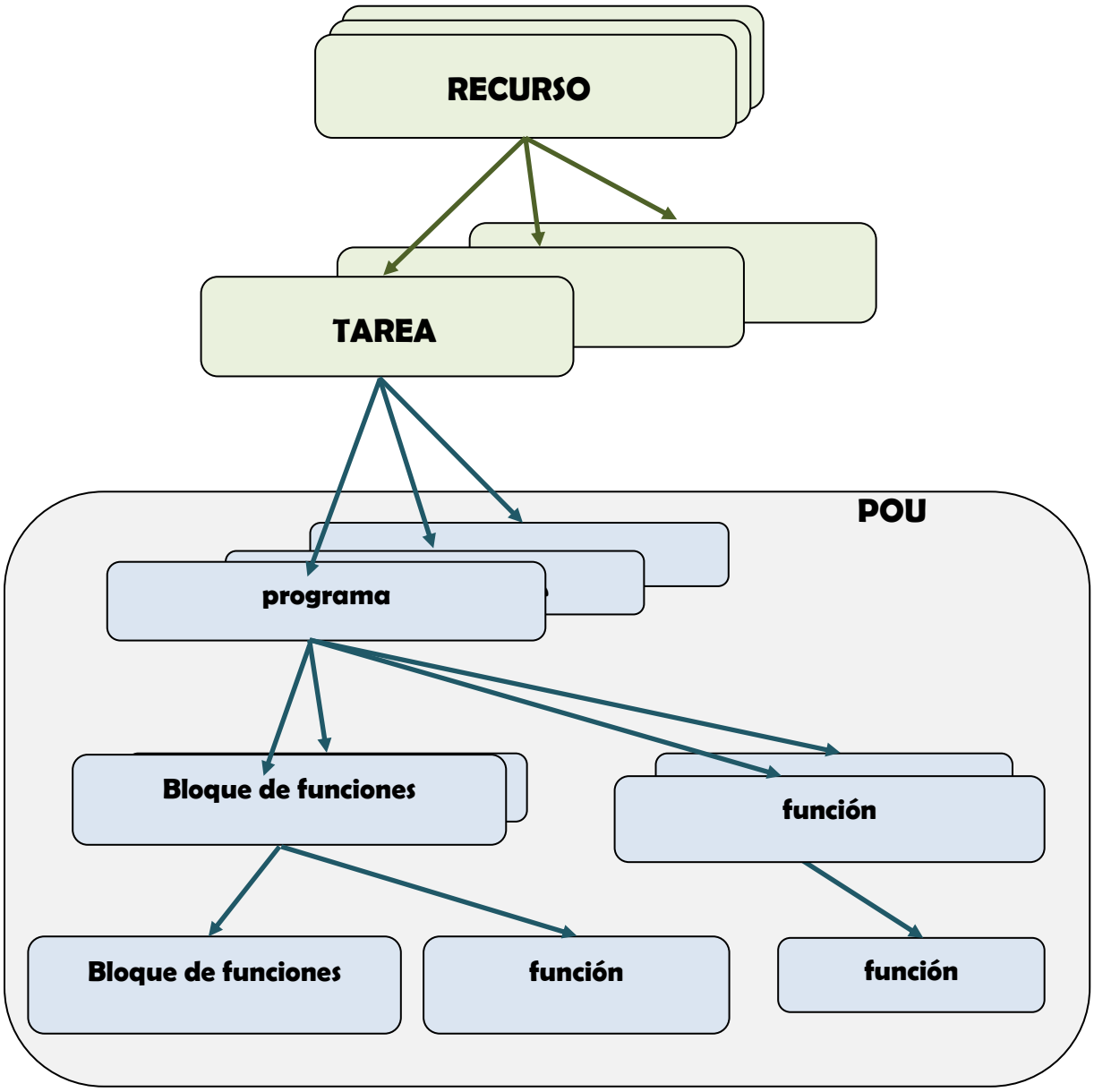
Un proyecto según la norma IEC 61131-3 consta de, al menos, una configuración. En cada configuración, encontramos uno o más recursos. Un recurso contiene todas las herramientas necesarias para ejecutar un programa determinado de usuario. Es una característica para mantener un determinado procesador.
Podemos entender mejor una configuración como el bastidor de un PLC y un recurso como una unidad central (CPU) con el sistema operativo incluido.
Pueden estar activas varias tareas en cada recurso. Se crea una tarea para ejecutar varios programas. Una tarea o TASK puede ser una cíclica (periódica), una tarea activada por un evento o una tarea activada por un sistema (interrupción).

Una tarea cíclica se ejecuta de forma cíclica con un control del tiempo.

Una tarea activada por un evento se ejectua en el marco de un acto particular. En el PC WorX, la tarea de evento se utiliza para la posibilidad de un tratamiento secuencial, en cada ciclo de INTERBUS.

Una tarea activada por un sistema se crea por el sistema operativo (arranque en caliente, arranque en templado, parada de la CPU, error de watchdog, división por cero, ...).
El programa de ejecución que encontramos en una tarea puede estar compuesto de diferentes POU (Program Organisation Unit o Unidades de Organización del Programa).

POU es la denominación colectiva para los programas, los bloques de funciones o las funciones.



5.3 POU

POU significa Unidades de Organización del Programa (Program Organisation Units) y es un nombre colectivo para los programas, los bloques de funciones y las funciones.

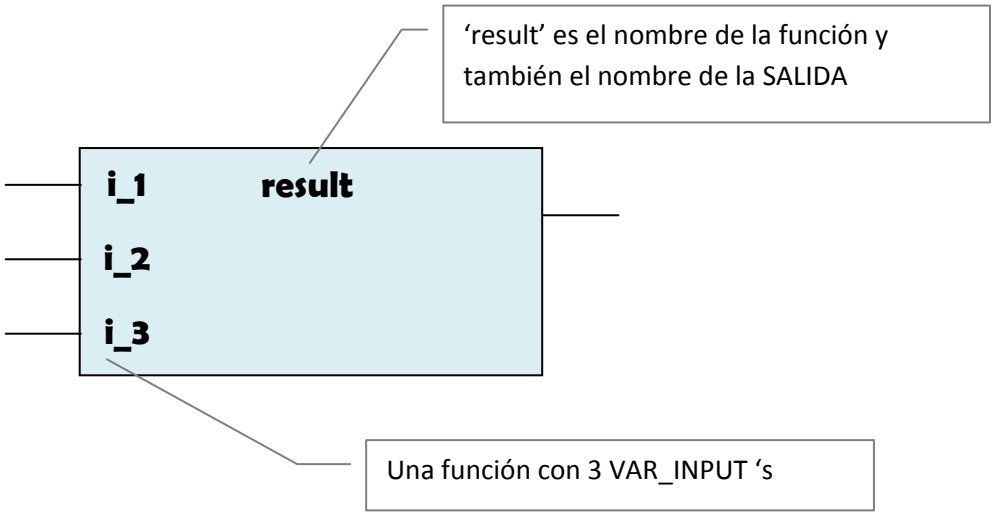
Resumen :

Tipo de POU	Utilizado algunas veces como	Comentario
Programa	Programa	Los elementos software reutilizables más importantes en un programa de usuario
Bloque de funciones	Bloque función	Los elementos software reutilizables menos importantes (desde algoritmos complejos a tareas de control simples)
Función	Función	Los elementos software reutilizables en la manipulación de datos

5.3.1 Funciones

Generalidades
Una función es un bloque software que se puede utilizar varias veces en otro POU. Casi siempre, las funciones se utilizarán para manipular datos. Una función produce siempre, un único resultado partiendo de varios parámetros de entrada.
El resultado se atribuye al nombre de la propia función. Para realizarlo, una función tendrá que ser siempre de un tipo de datos determinado.

- ✓ Una función tiene sólo una salida.
- ✓ Para un conjunto idéntico de parámetros de entrada, una función dará siempre el mismo resultado.



Variables y funciones
Una variable en una función puede ser del tipo VAR y VAR_INPUT. Las variables del tipo VAR_OUTPUT no son posibles porque una función sólo tiene una salida, a saber, el nombre de la propia función.

Funciones estándar
Existe un gran número de funciones estándar. Destacaremos los siguientes grupos :

- **funciones numéricas** (ABS, SQRT, LN, LOG, EXP, SIN, COS, TAN, ASIN, ACOS, ATAN).
- **funciones aritméticas** (ADD, MUL, SUB, DIV, MOD, EXPT, MOVE).
  Estas funciones son casi siempre del tipo de datos ANY_NUM. Esto significa que el tipo de datos de los parámetros de entrada puede ser tanto INT, DINT como REAL.
- **funciones bit string** (SHL, SHR, ROR, ROL).
- **Funciones Booleanas** (AND, OR, XOR, NOT).
  Las funciones de los dos últimos grupos son casi siempre del tipo de datos ANY_BIT. Esto significa que el tipo de datos de los parámetros de entrada puede ser tanto BIT, BYTE, WORD, como una palabra compuesta.
- **funciones de comparación** (GT, GE, EQ, LE, LT, NE).
- **funciones string caracteres** (LEFT, RIGHT, MID, CONCAT, INSERT, DELETE, REPLACE, LEN, FIND) funciones para convertir tipos de datos.

Para PC WorX, existen además numerosas bibliotecas disponibles con funciones especiales.

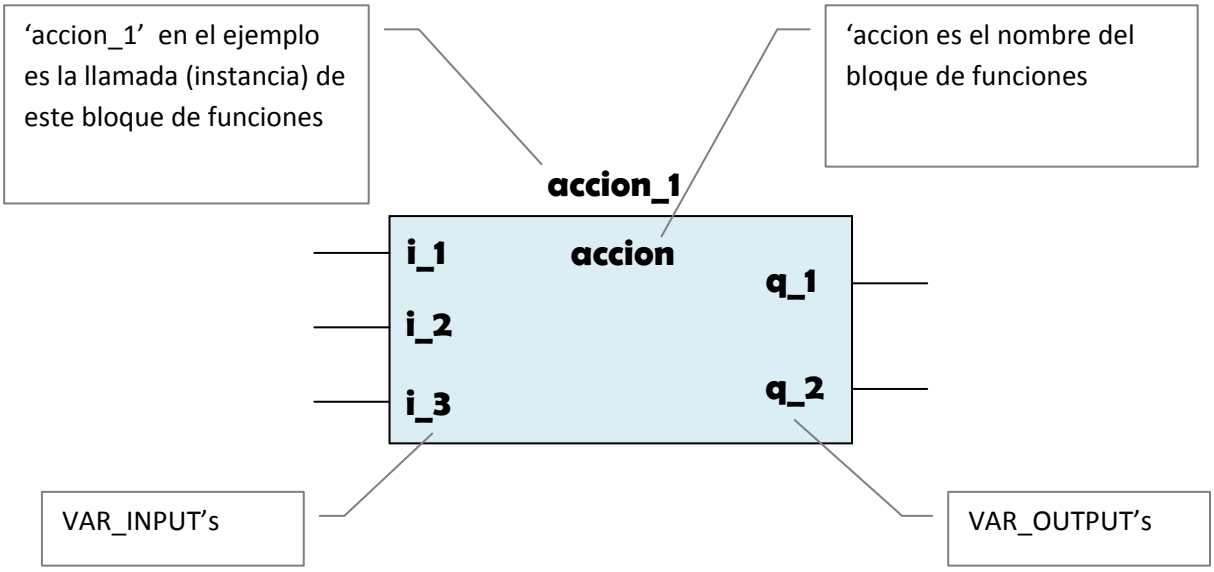
5.3.2 Bloques de funciones

Generalidades
Ejemplos de bloques de funciones : Un bloque de funciones es un elemento software reutilizable para ejecutar tareas de mando desde complejas a simples. Cuando se crea un bloque de funciones, se definen las variables. Desde la llamada a este bloque de funciones, se reserva un espacio memoria para el mismo. Si se quiere llamar varias veces a este bloque de funciones, hay que prever varias veces un espacio memoria. Es la razón por la cual se define una instancia cuando se llama a un bloque de funciones. Esta instancia reenvía el bloque de funciones a este emplazamiento de memoria específica.
En el caso de un conjunto determinado de parámetros de entrada, los parámetros de salida de un bloque de funciones podrán tener un resultado diferente en función del tiempo. Es decir, el bloque de funciones puede tener memoria interna.

Variables y bloques de funciones
Una variable en un bloque de funciones puede ser del tipo VAR, VAR_GLOBAL, VAR_INPUT y VAR_OUTPUT. A este nivel, los parámetros de entrada son del tipo VAR_INPUT.

Los valores de salida del bloque de funciones son del tipo VAR_OUTPUT.
Una variable puede también ser del tipo VAR_IN_OUT. Las variables de este género se ven atribuir un valor de entrada que puede cambiar en la ejecución del bloque de funciones.

Se debe evitar la utilización de VAR_GLOBAL en un bloque de funciones.



Bloques de funciones estándar
Hay muchos bloques de funciones definidos previamente. Los principales son:

- SR, RS, R_TRIG, F_TRIG,
- **Contadores :** CTU, CTD, CTUD
- **Temporizadores :** TP,TOF, TON

5.3.3 Programas

Generalidades
Un programa es un elemento software reutilizable importante. Este elemento se compone casi siempre de otros elementos software, como las funciones y los bloques de funciones. Estos elementos se escriben en uno de los 5 lenguajes IEC.
Un programa típico se compone de varios bloques de funciones relacionados que pueden intercambiar datos mediante el software.

Variables
Las variables pueden ser del tipo VAR, VAR EXTERNAL, VAR_EXTERNAL_PG y VAR_GLOBAL.

Las variables globales se utilizan en los programas y, a continuación, se asocian a una entrada–salida del bus de campo (INTERBUS o PROFINET) o a variables internas.

www.infoPLC.net

5.4 Tipos de datos

5.4.1 Tipos generales de datos

En los cuadros siguientes, damos un resumen de los principales tipos de datos.

Tipo de dato	Descripción	Bits	Capacidad
SINT	Short integer (entero corto)	8	de -128 a +127
INT	Integer (entero)	16	de -32768 a 32767
DINT	Double integer (entero doble)	32	de -2^{31} a $+2^{31}-1$
LINT	Long integer (entero largo)	64	de -2^{63} a $+2^{61}-1$
USINT	Unsigned short integer (entero corto sin signo)	8	de 0 a 255
UINT	Unsigned integer	16	de 0 a $2^{16}-1$
UDINT	Unsigned double integer (entero doble sin signo)	32	de 0 a $2^{32}-1$
ULINT	Unsigned long integer (entero largo sin signo)	64	de 0 a $2^{64}-1$
REAL	Floating point (real en coma flotante)	32	$\pm 10^{\pm 38}$
LREAL	Long floating point (real en coma flotante largo)	64	$\pm 10^{\pm 308}$

- Observaciones sobre la notación de los tipos de datos numéricos
  INT#15 o 15
  REAL#8 o 8.0

Tipo de dato	Descripción	Bits	Capacidad
BOOL	1 bit	1	1 o 0 lógico
BYTE	Bitstring de 8 bits	8	Informaciones binarias
WORD	Bitstring de 16 bits	16	Informaciones binarias
DWORD	Bitstring de 32 bits	32	Informaciones binarias
LWORD	Bitstring de 64 bits	64	Informaciones binarias
STRING	Cadena de caracteres		Informaciones textuales

- Observaciones referentes a la notación de los tipos de datos basados en los bits

  WORD#2#1010111100000011 (notación binaria)
  WORD#16#AF03 (notación hexadecimal)
  WORD#44803 (notación decimal)
  BYTE#2#10100011

  El mismo valor en notación binaria, hexadecimal y decimal

Tipo de dato	Descripción	Bits	Capacidad
TIME	Duración en el tiempo		
DATE	Datos del calendario		
TIME_OF_DAY	Hora del día		
DATE_AND_TIME	Datos del calendario + hora del día		

- Observaciones sobre TIME
  T# 3s
  T# 16d_5h_3m_4s_20ms (16 días, 5 horas, 3 minutos, 4 segundos y 20 milésimas de segundo)
  TIME# 10s

- Observaciones sobre DATE
  D#2001-10-15 (15 octubre 2001)
  DATE#1999-1-30

- Observaciones relativas a TIME_OF_DAY
  TOD#10 : 10:30 (10 horas, 10 minutos y 30 segundos)
  TIME_OF_DAY#23 :59:59 (1 segundo antes de medianoche)

- Observaciones respecto a DATE_AND_TIME
  DT#2007-03-22-15:36 :50
  DATE_AND_TIME# 2007-02-01-12:00:00

5.4.2 Matriz (Array)

El tipo de datos « Array », o matriz, es una estructura de varios elementos del mismo tipo de datos.

Los diferentes elementos se pueden manipular separadamente mediante un índice array.

Un tipo de datos array se ha de declarar previamente. Cuando se declara el tipo de datos, se pueden definir, a continuación, las variables de este tipo de datos array.

```
TYPE
    (*Typename*) : ARRAY [(*From..To*)] OF (*DATATYPE*);
END_TYPE
```

Ejemplo de un array unidimensional

Declaración del tipo de datos array

```
TYPE
   numeros_serie : ARRAY[1..23] OF INT;
END_TYPE
```

Definición de las variables

```
VAR
   serie_1 : numeros_serie;
   valor_4: INT;
END_VAR
```

Código en ST

```
valor_4 := serie_1 [4]
```

Ejemplo de un array multidimensional

Declaración del tipo de datos array

```
TYPE
   numeros_serie : ARRAY[1..23] OF INT;
   resultados : ARRAY [1..3] OF numeros_serie;
END_TYPE
```

Definición de las variables

```
VAR
   linea_produccion_1 : resultados;
   valido_x: INT;
END_VAR
```

Código en ST

```
valido_x := linea_produccion_1 [2] [10];
```

www.infoPLC.net

5.4.3 Estructura (Struct)

El tipo de datos « Struct » es una estructura de diferentes elementos de diferentes tipos de datos.

```
TYPE
    (*Typename*) :          :
    STRUCT
        (*Element 1 Name*)   :      (*DATATYPE*);
        (*Element 2 Name*)   :      (*DATATYPE*);
        (*Element 3 Name*)   :      (*DATATYPE*);
        (*    .          :        .  *)
        (*    .          :        .  *)
        (*Element n Name*)   :      (*DATATYPE*);
    END_STRUCT;
END_TYPE
```

5.4.4 Cadena (String)

Un string, o cadena, es un tipo de datos para tratar el texto. Es un conjunto de bytes, donde cada byte representa un carácter (cuadro ASCII).

```
TIPO
    (*Typename*):STRING((*String length*));
END_TYPE
```

5.5 Texto Estructurado (Structured Text)

Las especificidades del textp estructurado son las declaraciones condicionales. Aquí se presenta un breve resumen.

5.5.1 Selección de declaraciones

IF… THEN… ELSE…

SI..ENTONCES… SI NO….

```
IF      <expresión booleana>      THEN
          <declaración>;
ELSE
          <declaración>;
END_IF;
```

Con ELSIF, se convierte en :

```
IF      < expresión booleana >    THEN
          <declaración>;
ELSIF  < expresión booleana >    THEN
          <declaración>;
ELSE
          <declaración>;
END_IF;
```

CASE

CASO

```
CASE      <expresión entera >     OF
<valor entero_1> :  <declaración>;
< valor entero_2> :  <declaración>;
< valor entero_3> :  <declaración>;

ELSE
          <declaración>;
END_CASE;
```

www.infoPLC.net

5.5.2 Declaraciones de iteración

FOR… DO PARA… HACER

```
FOR  <valor inicial>  TO  <valor final>  BY  <valor incremental >
        <declaración>;
       <declaración>;
        <declaración>;
END_FOR;
```

WHILE… DO MIENTRAS… HACER

```
WHILE  <expresión booleana>  DO
        <declaración>;
       <declaración>;
        <declaración>;
END_WHILE;
```

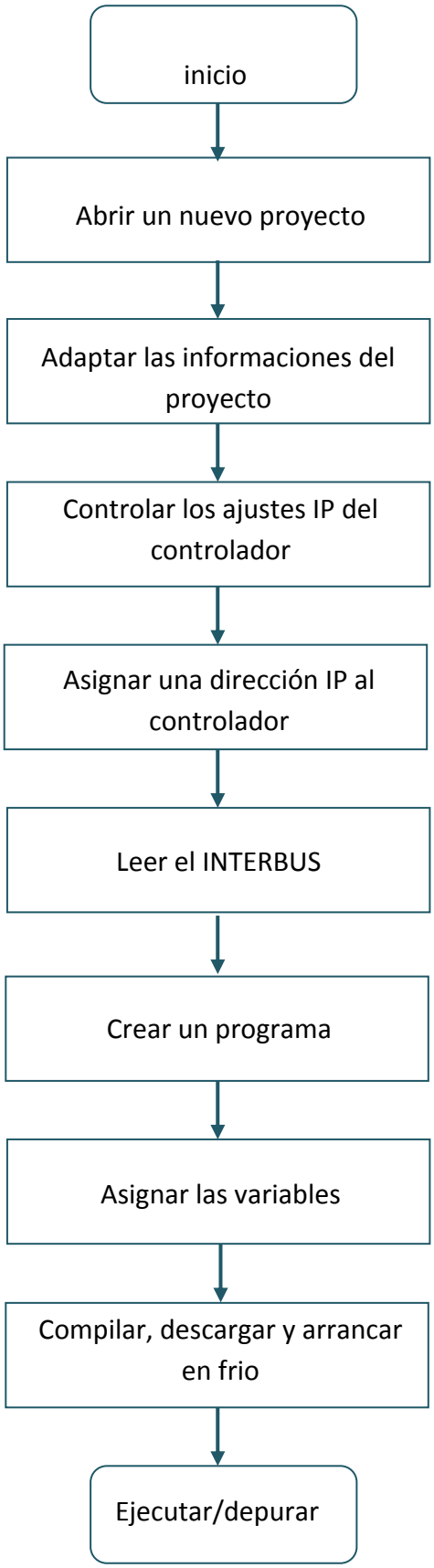
REPEAT… UNTIL REPETIR… HASTA

```
REPEAT
        <declaración>;
       <declaración>;
        <declaración>;
UNTIL   <expresión booleana>
END_REPEAT;
```

Con la declaración ***EXIT***, se puede dar un final condicional a una declaración de iteración.

6 Creación de un proyecto

6.1 Resumen



6.2 Abrir un nuevo proyecto

- Seleccione “Nuevo proyecto” en el menú “Archivo” para crear un nuevo proyecto mediante una plantilla. La estructura del proyecto, así como la posibilidad de elección de un controlador, están ahora preparadas.

- Seleccione el controlador “ILC 150 ETH Rev. >01/2.00” y confirme con Aceptar.

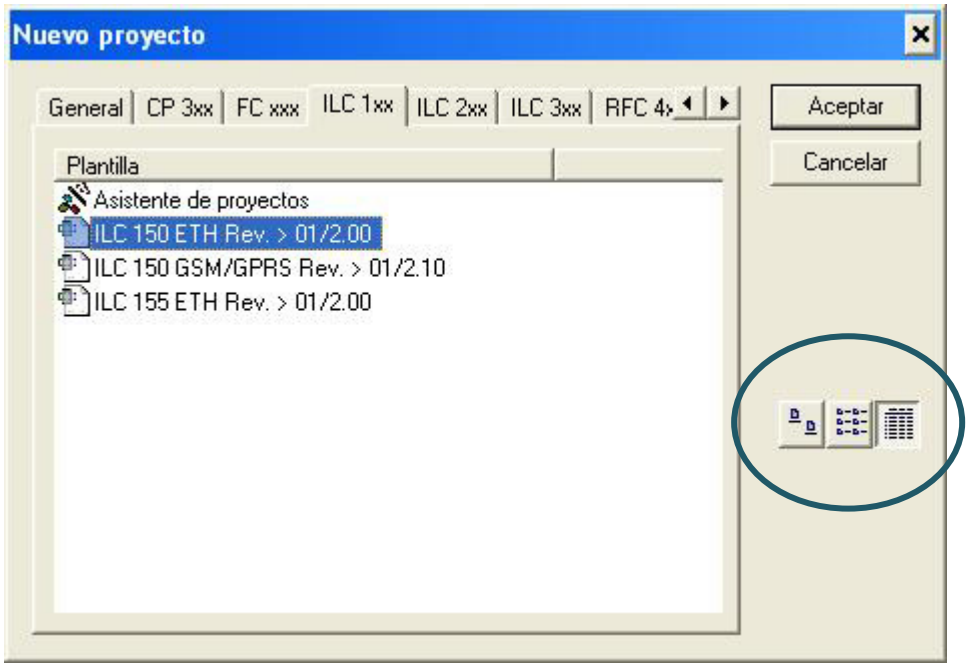


Ilustración 6-1 Plantillas de proyecto.

- Seleccione “Guardar proyecto como…” en el menú “Archivo”. Elija un nombre de proyecto. En función de la utilización de Webvisit,para realizar la visualización basada en web, no se permiten los espacios en el nombre (aquí QuickStart_ILC150, ver Ilustración 6-2) y archive el proyecto mediante “Guardar”.

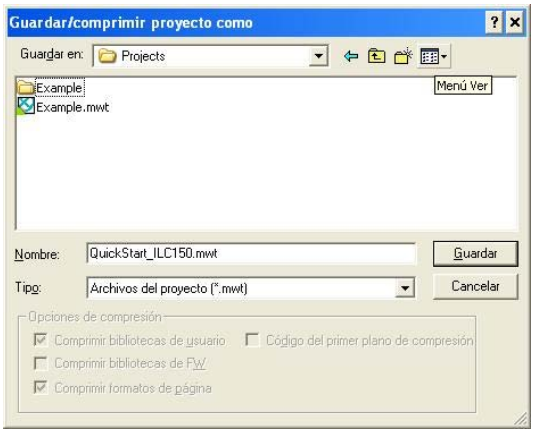


Ilustración 6-2

Cada proyecto se registra como un fichero *.mwt y una carpeta del mismo nombre. En la Ilustración 6-2, tenemos un fichero *.mwt con el nombre EXAMPLE, así como una carpeta con el nombre EXAMPLE.

También se puede comprimir el proyecto para obtener un archivo *.zwt.

6.3 Estructura del nuevo proyecto de acuerdo con la norma IEC 61131-3

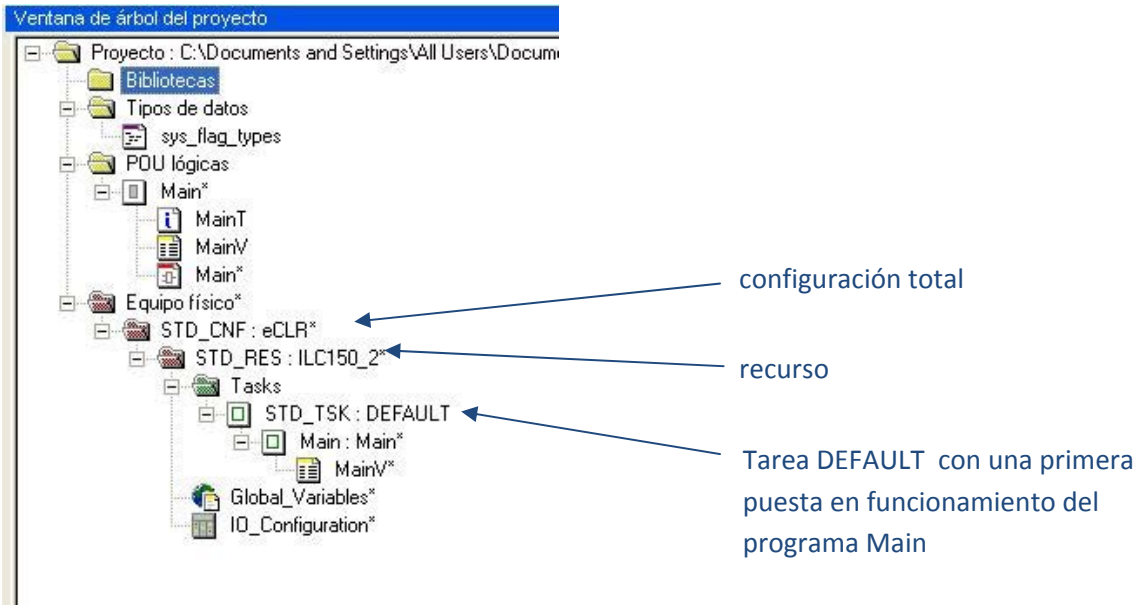


Ilustración 6-3 Estructura durante la creación de un nuevo proyecto en PC Worx.

Se ha creado ya un programa con el nombre Main en la carpeta 'POU lógicas'.

Este programa se utiliza la primera vez a través de la tarea DEFAULT.

Utilizando esta estructura, este programa se puede compilar inmediatamente.

Si se crean otros programas, se han de asignar a una tarea (ver más adelante en este curso).

6.4 Adaptación de las informaciones del proyecto

- Pase al área de trabajo “Configuración del Bus”. Se encontrará con todas las informaciones relativas al proyecto.

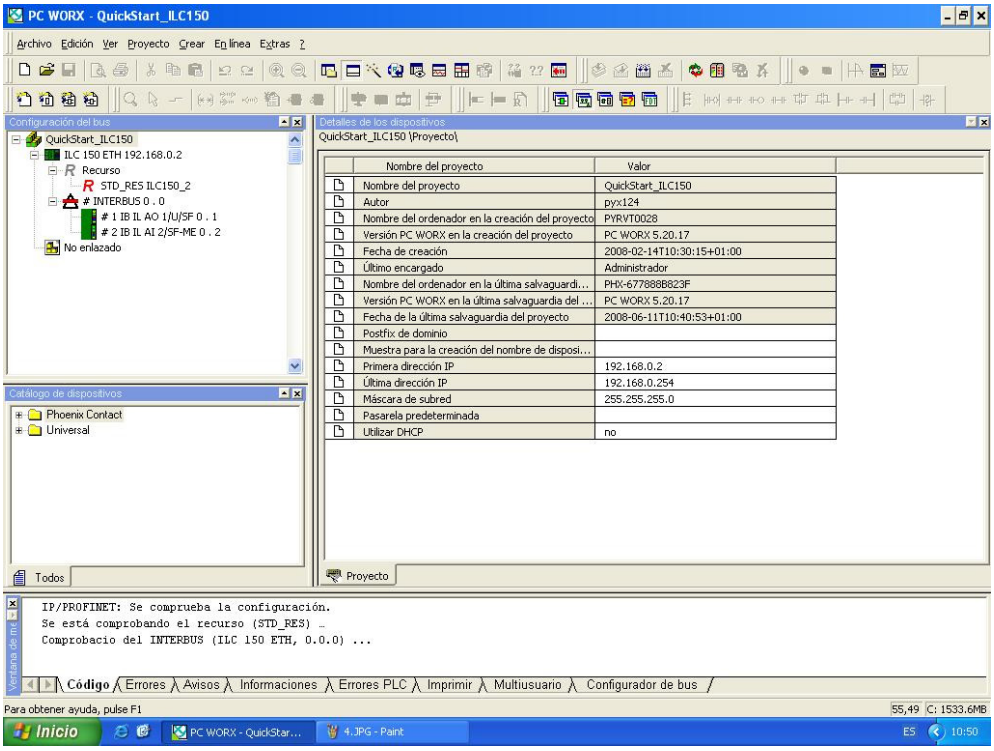


Ilustración 6-4

Durante la creación del proyecto, PC WorX atribuye automáticamente un rango de direcciones IP en una red local (192.168.0.2 a 192.168.0.254).

Si se ha de utilizar otra dirección, la dirección de partida y la dirección de finalización se han de adaptar ; si es necesario, también hay que adaptar la máscara de subred.

Para el curso, se aplican las siguientes direcciones IP :	
Ordenador	192.168.0.10
ILC_150	192.168.0.11

Ver también el plano de situación de la página siguiente.

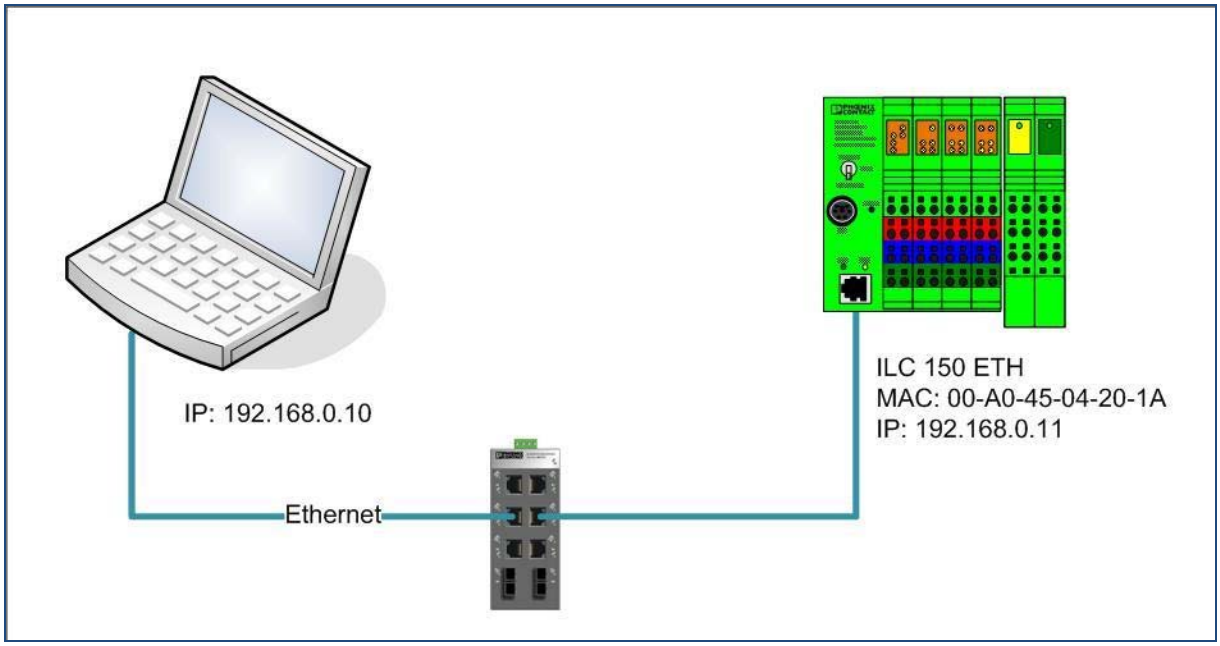


Ilustración 6-5 - Configuración.

6.5 Configuración IP del ordenador

Es importante controlar primero los ajustes de red del ordenador.
Abra una sesión de DOS para solicitar la dirección IP del ordenador mediante “ipconfig”.

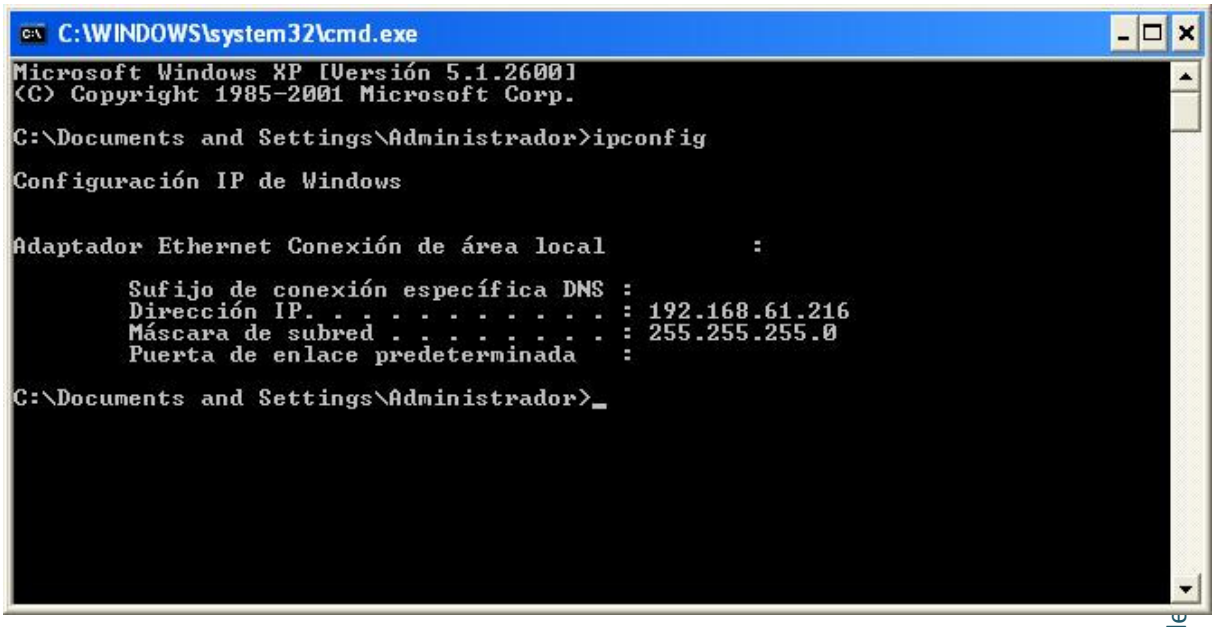

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrador>ipconfig
Configuración IP de Windows
Adaptador Ethernet Conexión de área local :
Sufijo de conexión específica DNS :
Dirección IP. . . . . : 192.168.61.216
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada :
C:\Documents and Settings\Administrador>

Ilustración 6-6 Ipconfig.

Adapte sus ajustes de ordenador, si es necesario, mediante las propiedades de la conexión LAN. A nivel de protocolo Internet, seleccione las propiedades y ajuste una dirección IP fija.

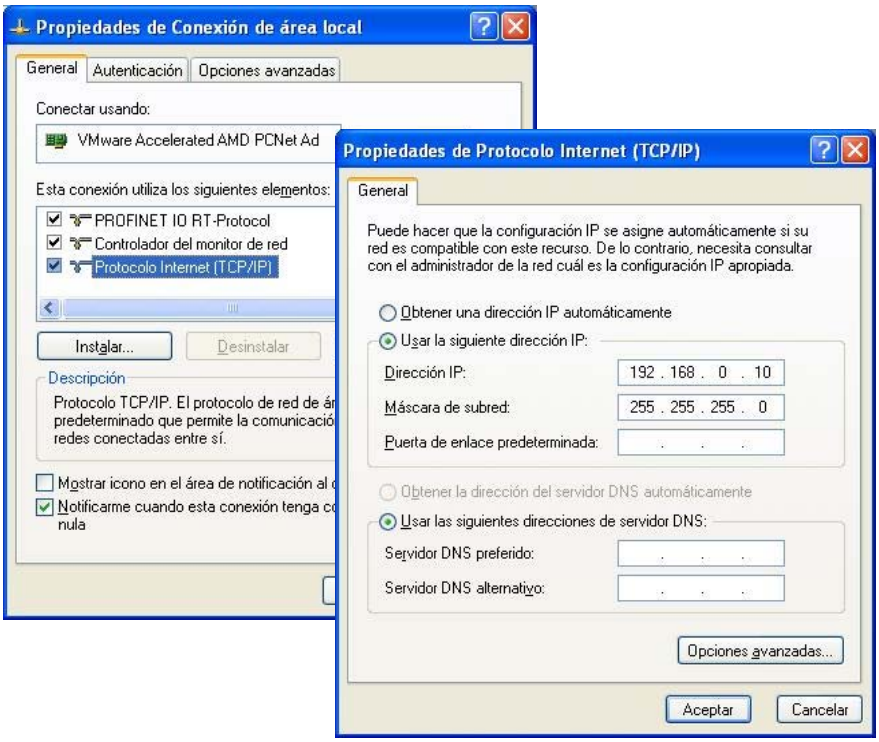

Propiedades de Conexión de área local
General
Autenticación
Opciones avanzadas
Conectar usando:
VMware Accelerated AMD PCNet Ad
Esta conexión utiliza los siguientes elementos:
PROFINET IO RT-Protocol
Controlador del monitor de red
Protocolo Internet (TCP/IP)
Instalar...
Desinstalar
Descripción
Protocolo TCP/IP. El protocolo de red de ár
predeterminado que permite la comunicació
redes conectadas entre sí.
Mostrar icono en el área de notificación al
Notificarme cuando esta conexión tenga c
nula
Propiedades de Protocolo Internet (TCP/IP)
General
Puede hacer que la configuración IP se asigne automáticamente si su red es compatible con este recurso. De lo contrario, necesita consultar con el administrador de la red cuál es la configuración IP apropiada.
Obtener una dirección IP automáticamente
Usar la siguiente dirección IP:
Dirección IP: 192 . 168 . 0 . 10
Máscara de subred: 255 . 255 . 255 . 0
Puerta de enlace predeterminada:
Obtener la dirección del servidor DNS automáticamente
Usar las siguientes direcciones de servidor DNS:
Servidor DNS preferido:
Servidor DNS alternativo:
Opciones avanzadas...
Aceptar
Cancelar

Ilustración 6-7

6.6 Configuración IP del controlador

6.6.1 Generalidades

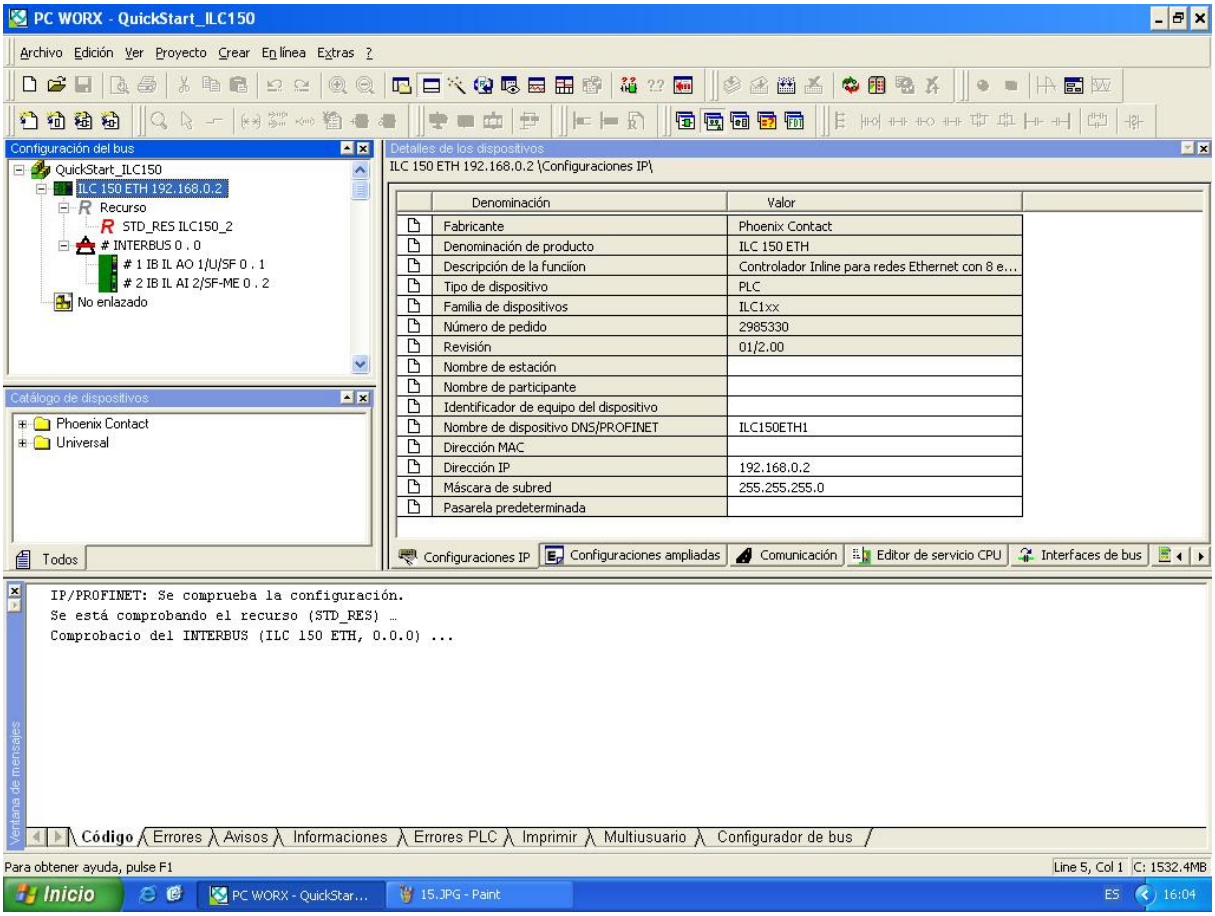


Ilustración 6-8

Para adaptar la dirección IP, existen las siguientes posibilidades :

- Asignar la primera vez una dirección IP
  - Mediante un servidor BootP
  - Mediante el interfaz serie

- Modificar la dirección IP
  - Mediante el inferfaz serie
  - Mediante Ethernet con PC WorX

- ¿Ha olvidado la dirección IP ?
  - Solicitarla mediante BootP Server

6.6.2 Adaptar la dirección IP mediante BootP Server

Ilustración 6-9 BootP.

- Active la casilla marcada “Servidor BootP activo”.
- Pase al área de trabajo “Configuración del Bus”.
- Seleccione el controlador “ILC 150 ETH".
- Seleccione la pestaña “Configuraciones IP” de la ventana “Detalles de los dispositivos”.
- Introduzca la dirección MAC del controlador (la dirección MAC empieza por 00.A0.45).
- Complete la dirección IP deseada.

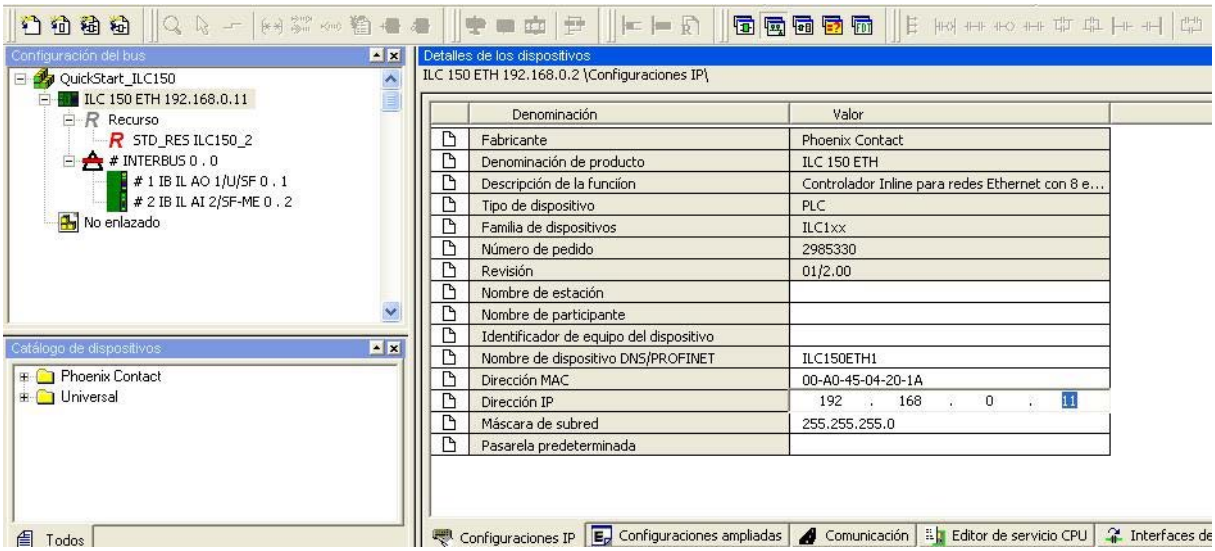


Ilustración 6-10

- Pulse el botón de reinicio del controlador.

- Suelte el botón de reinicio. A partir de ahora, se ha lanzado el proceso BootP, el controlador recibe la dirección IP introducida y después de unos segundos, se activa la dirección IP.

Una vez que se introduce la dirección IP correcta en el controlador y en el ordenador, se puede activar la comunicación.

Para realizarlo, hay que clicar en ILC 150 ETH, en la ventana “Configuración del bus” (a la izquierda) y seleccionar la pestaña “Comunicación” en la ventana “Detalles de los dispositivos”.

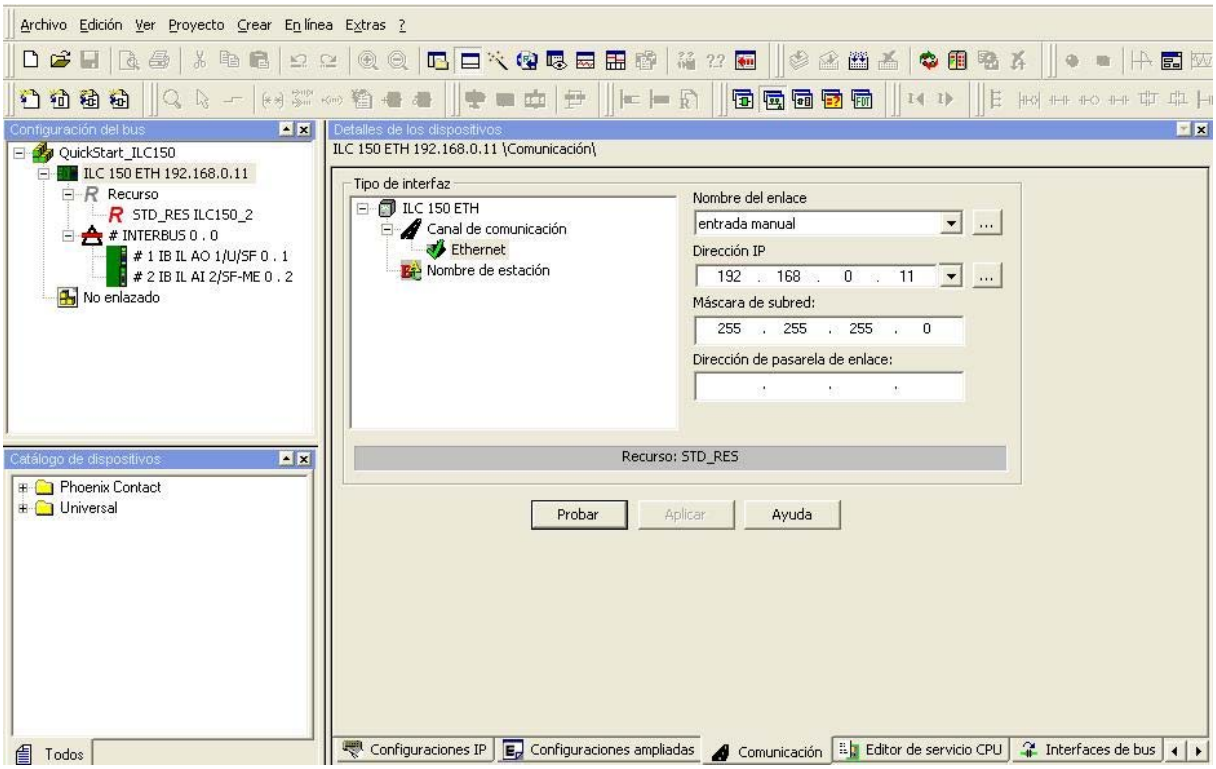


Ilustración 6-11

Después de introducir la dirección IP correcta del controlador, se puede comprobar la comunicación mediante el botón “Probar”.

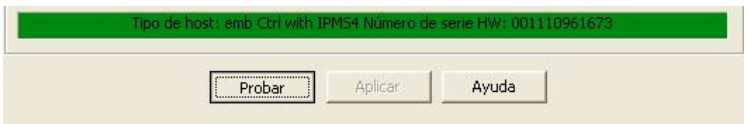


Ilustración 6-12

Tras unos segundos, se encenderá en verde la barra de indicación y se podrá confirmar la comunicación mediante el botón "Aplicar".

6.7 La configuración INTERBUS

Desde que se ha activado el camino de comunicación entre el controlador y el PC, se puede leer automáticamente el sistema INTERBUS asociado al controlador.

Para este proyecto, se han asociado 2 módulos INTERBUS al controlador.

- Seleccione “Bus conectado” mediante el menú “Ver” para leer el sistema INTERBUS (Ilustración 5-5).

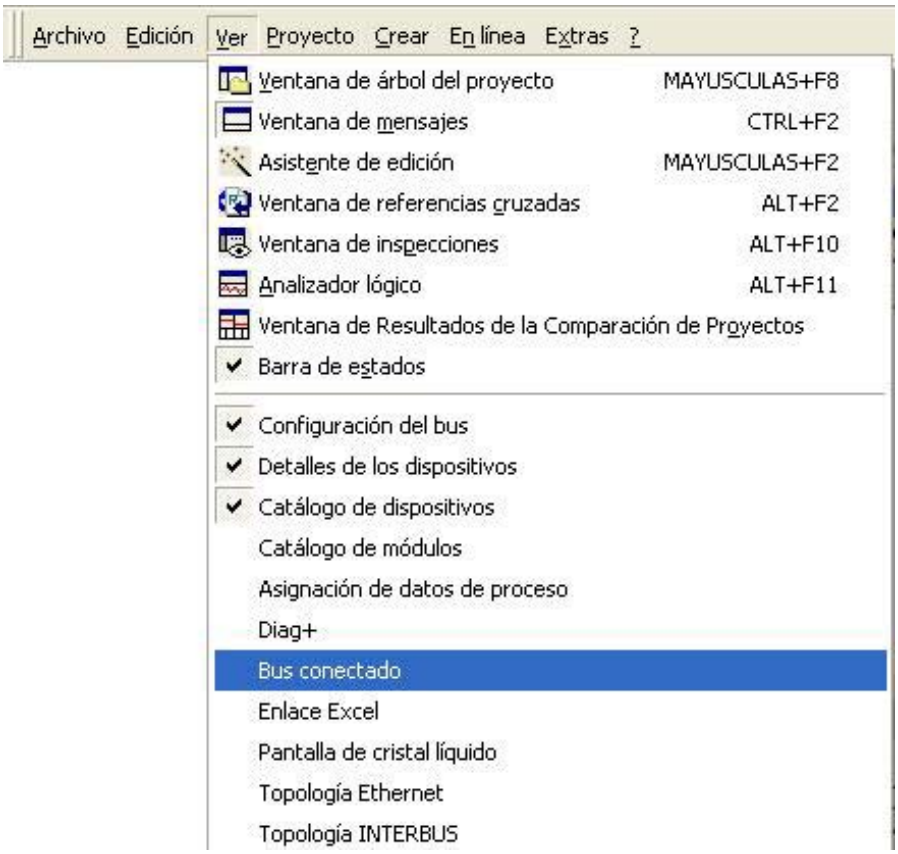


Ilustración 6-13

- Seleccione el controlador correcto en la ventana “Bus conectado”.

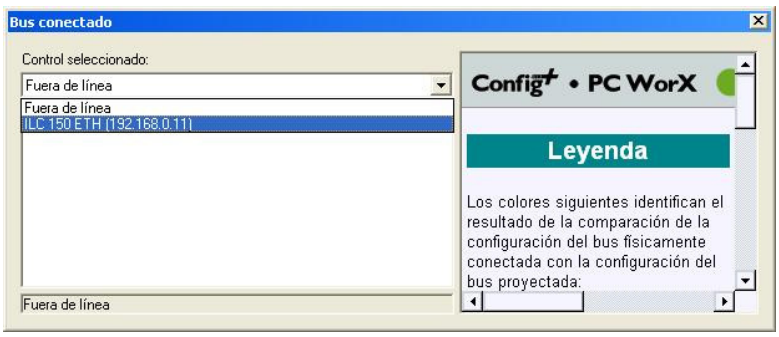

Bus conectado
Control seleccionado:
Fuera de línea
Fuera de línea
ILC 150 ETH (192.168.0.11)
Fuera de línea
Config+ • PC WorX
Leyenda
Los colores siguientes identifican el resultado de la comparación de la configuración del bus físicamente conectada con la configuración del bus proyectada:

Ilustración 6-14

- Los módulos INTERBUS se leen ahora automáticamente. Una vez que se leen, se han de añadir al proyecto.
- Para realizarlo, pulse sobre el controlador, con el botón derecho del ratón y seleccione, a continuación, “Aceptar en el proyecto” y “Con descripción de dispositivo”.

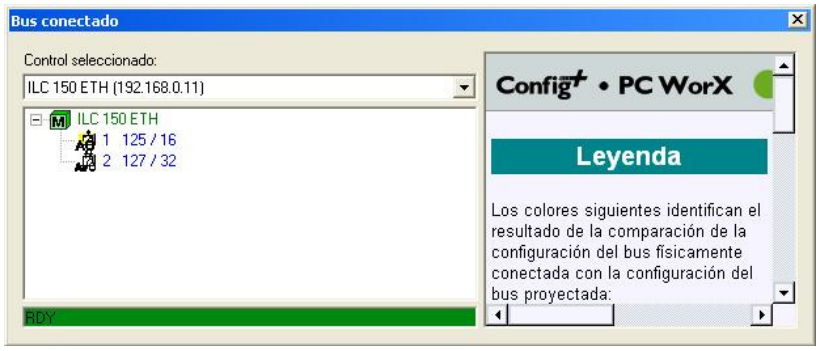

Bus conectado
Control seleccionado:
ILC 150 ETH (192.168.0.11)
ILC 150 ETH
1 125 / 16
2 127 / 32
RDY
Config+ • PC WorX
Leyenda
Los colores siguientes identifican el resultado de la comparación de la configuración del bus físicamente conectada con la configuración del bus proyectada:

Ilustración 6-15

- La ventana “Seleccionar dispositivo” presenta ahora todos los módulos que corresponden al código ID (de identificación) de los módulos conectados. Se han de seleccionar de esta lista los módulos realmente asociados.

Los módulos INTERBUS utilizados en este proyecto :

- IB IL AO 1/U/SF
- IB IL AI 2/SF-ME

- Complete ahora la configuración, como en la Ilustración 6-16 :

Configuración del bus
QuickStart_ILC150
ILC 150 ETH 192.168.0.11
Recurso
STD_RES ILC150_2
INTERBUS 0 . 0
1 IB IL AO 1/U/SF 0 . 1
2 IB IL AI 2/SF-ME 0 . 2
No enlazado

Ilustración 6-16

- Para acabar, es mejor seleccionar fuera de línea ahora, antes de cerrar esta ventana.

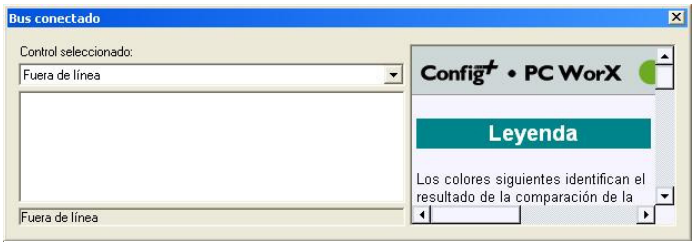


Ilustración 6-17

6.8 Compilación y descarga

El proyecto contiene ahora todas las informaciones referentes a las configuraciones en bus (Ethernet & INTERBUS) y hay una comunicación activa. En este momento, es posible compilar el proyecto y descargarlo para identificar los eventuales errores.

compilar → descargar al controlador → ejecutar un arranque templado.

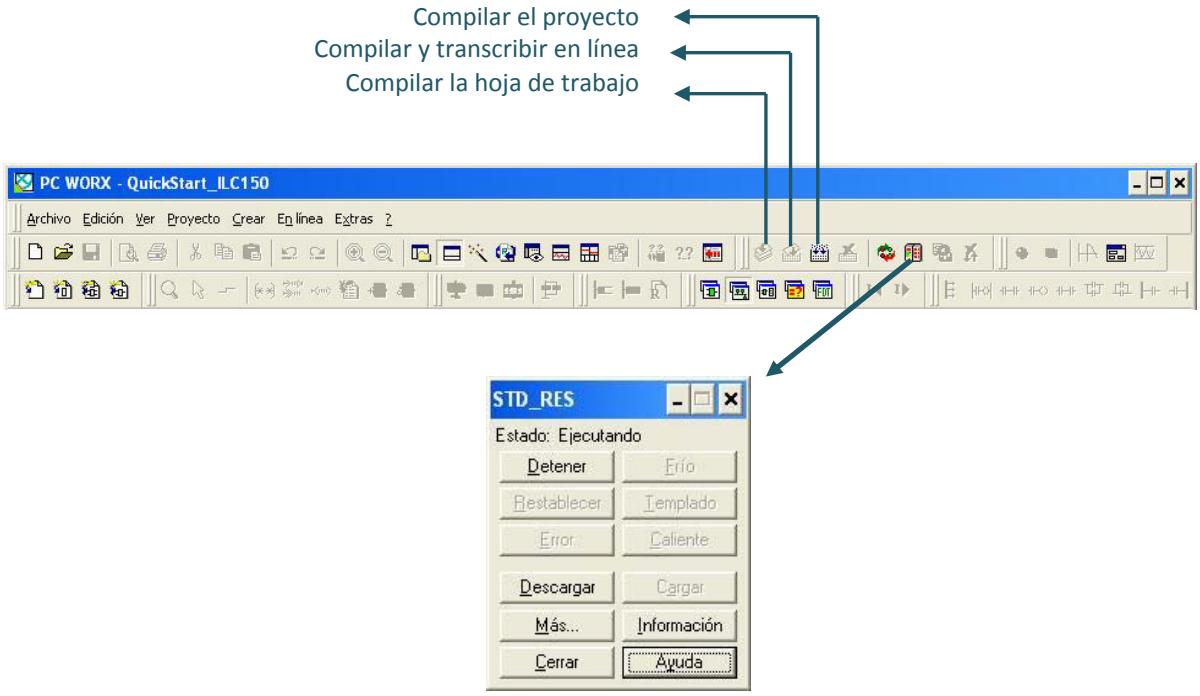


Ilustración 6-18

Algunas teclas de acceso rápido importantes relativas a la compilación :

Ilustración 6-19

www.infoPLC.net

7 Programa de ejemplo

7.1 Introducción

Encontrará más abajo un programa de ejemplo simple descrito paso a paso. El funcionamiento previsto comprende 2 partes :

1. **PC-WorX** : En PC WorX, se escribe un programa sencillo, que permite activar un parpadeo en una salida mediante un interruptor. Por otra parte, también se puede leer y visualizar en una barra LED un valor analógico.

2. **Webvisit :** El ILC150 ETH dispone de un servidor web. En el marco de este ejercicio, el objetivo es crear una página web con la cual se pueda visualizar otra vez el parpadeo, así como posibilitar su mando. También se visualizará aquí el estado de la entrada analógica.

7.2 Descripción de la misión

Después de una configuración correcta del hardware (bus de campo), se explicará el entorno de programación de PC WorX, mediante un programa ejemplo, de acuerdo con la norma IEC61131-3.

El programa ejemplo comprende la programación de un parpadeo simple, mediante el cual un interruptor de conmutación activa el parpadeo de una salida en el módulo de salida Inline. También se puede conseguir el control de este parpadeo mediante un panel de visualización y de Webvisit.

Una segunda función es la asociación de una entrada analógica (potenciómetro) hacia una salida analógica (Gráfico de barras LED) mediante algunas operaciones. El valor de la salida analógica también se podrá controlar mediante Webvisit.

7.3 Elaboración

En primer lugar, hay que pasar al área de trabajo ‘Programación IEC”. Se observa inmediatamente que PC WorX está constituido siguiendo la norma IEC 61131-3. En la “Ventana de árbol del proyecto”, encontramos unas subdivisiones claras. Hay una estructura de carpetas para las bibliotecas, para los tipos de datos, para las POU y está el equipo físico.

Esta estructura se muestra en la Ilustración 7-1.

La estructura principal se encuentra en “Equipo físico”. La STD_CNF (configuración estándar) consta de un STD_RES (recurso estándar) que aquí es el ILC 150 ETH.

Este recurso puede ejecutar las diferentes misiones. Al inicio de un nuevo proyecto, siempre hay un programa “Main” que está predefinido en la carpeta “POU lógicas” y que se activa en la tarea “Default”.

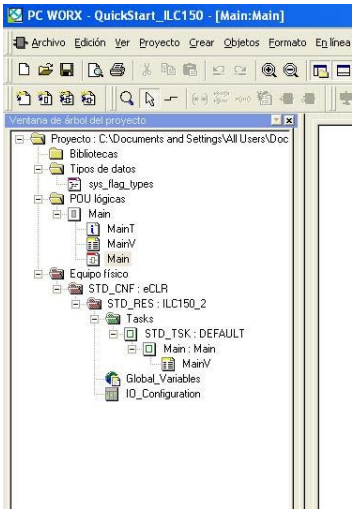


Ilustración 7-1

7.3.1 Creación de una nueva POU

Mediante los 4 iconos indicados, se puede añadir una nueva POU. Se puede elegir entre un nuevo programa (), una nueva función () o un nuevo bloque de funciones ().

En la ventana “Insertar”, hay que continuar especificando la POU.
Se ha de seleccionar un de los cinco lenguajes de programación (de acuerdo con la norma) para la POU.
Sólo en el caso de una función, se ha de seleccionar un “Tipo de datos del valor de retorno”.

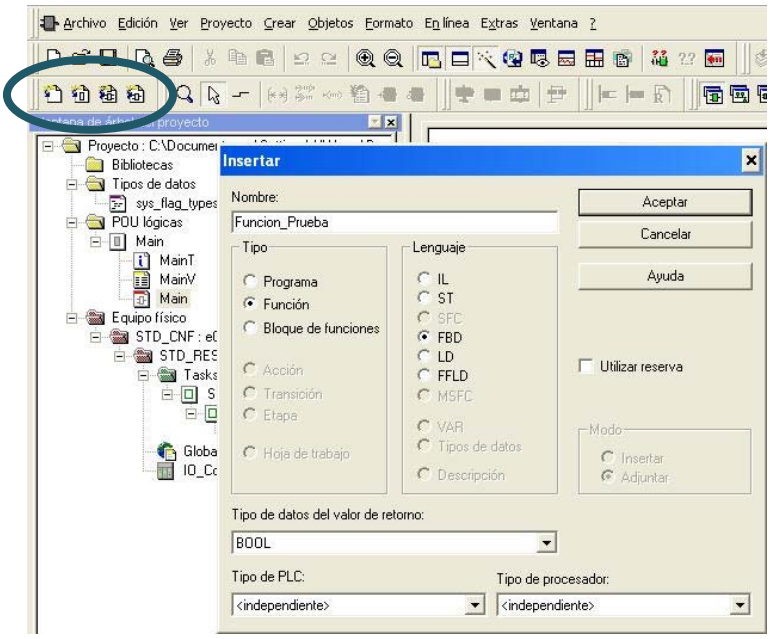


Ilustración 7-2 Creación de una nueva POU.

Para cada POU, se han creado tres hojas de trabajo:

- Nombre_POUT(); Campo de texto : todo lo que se anote aquí aparecerá en la solicitud de ayuda relativa a este POU.
- Nombre_POUV(): Campo para introducir variables.
- Nombre_POU(, , , ,): Campo de programación.

7.3.2 Definición de las variables

Hay que distinguir entre dos clases de variables.

- "Variables de usuario" : aquí, el nombre y el tipo se pueden elegir libremente.
- "Variables de sistema" : hay que buscar en la lista de variables de sistema existentes.

Las variables de sistema son siempre variables globales, lo que significa que estas variables se pueden solicitar o manipular en todo el proyecto (desde cualquier POU). Existen variables definidas de antemano que están activas en el PLC.

Encontrará una lista de todas las variables de sistema en el elemento "Global Variables" de la ventana de árbol de proyecto, ver ilustración siguiente :

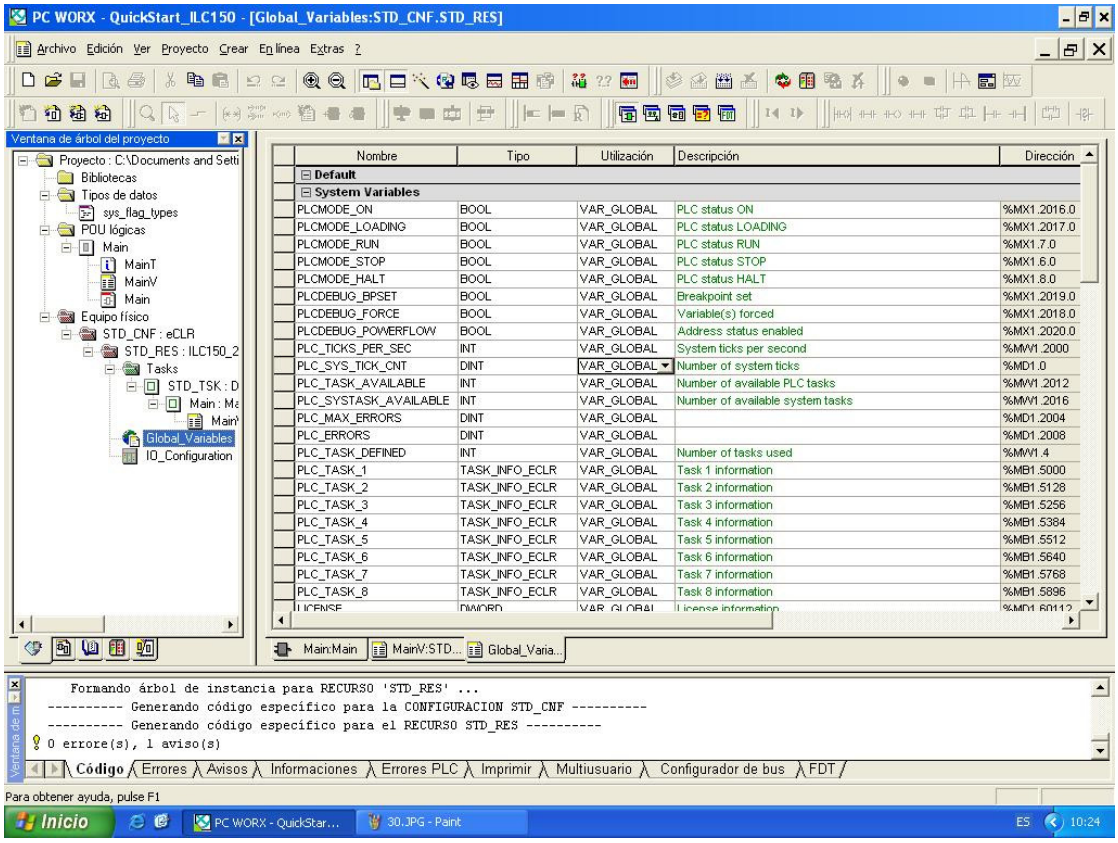

PC WORX - QuickStart_ILC150 - [Global_Variables:STD_CNF.STD_RES]
Archivo Edición Ver Proyecto Crear En línea Extras ?
Ventana de árbol del proyecto
Proyecto : C:\Documents and Setti
Bibliotecas
Tipos de datos
sys_flag_types
POU lógicas
Main
MainT
MainV
Main
Equipo físico
STD_CNF : eCLR
STD_RES : ILC150_2
Tasks
STD_TSK : D
Main : Ma
Main
Global_Variables
IO_Configuration
Nombre
Tipo
Utilización
Descripción
Dirección
Default
System Variables
PLCMODE_ON BOOL VAR_GLOBAL PLC status ON %MX1.2016.0
PLCMODE_LOADING BOOL VAR_GLOBAL PLC status LOADING %MX1.2017.0
PLCMODE_RUN BOOL VAR_GLOBAL PLC status RUN %MX1.7.0
PLCMODE_STOP BOOL VAR_GLOBAL PLC status STOP %MX1.6.0
PLCMODE_HALT BOOL VAR_GLOBAL PLC status HALT %MX1.8.0
PLCDEBUG_BPSET BOOL VAR_GLOBAL Breakpoint set %MX1.2019.0
PLCDEBUG_FORCE BOOL VAR_GLOBAL Variable(s) forced %MX1.2018.0
PLCDEBUG_POWERFLOW BOOL VAR_GLOBAL Address status enabled %MX1.2020.0
PLC_TICKS_PER_SEC INT VAR_GLOBAL System ticks per second %MW1.2000
PLC_SYS_TICK_CNT DINT VAR_GLOBAL Number of system ticks %MD1.0
PLC_TASK_AVAILABLE INT VAR_GLOBAL Number of available PLC tasks %MW1.2012
PLC_SYSTASK_AVAILABLE INT VAR_GLOBAL Number of available system tasks %MW1.2016
PLC_MAX_ERRORS DINT VAR_GLOBAL %MD1.2004
PLC_ERRORS DINT VAR_GLOBAL %MD1.2008
PLC_TASK_DEFINED INT VAR_GLOBAL Number of tasks used %MW1.4
PLC_TASK_1 TASK_INFO_ECLR VAR_GLOBAL Task 1 information %MB1.5000
PLC_TASK_2 TASK_INFO_ECLR VAR_GLOBAL Task 2 information %MB1.5128
PLC_TASK_3 TASK_INFO_ECLR VAR_GLOBAL Task 3 information %MB1.5256
PLC_TASK_4 TASK_INFO_ECLR VAR_GLOBAL Task 4 information %MB1.5384
PLC_TASK_5 TASK_INFO_ECLR VAR_GLOBAL Task 5 information %MB1.5512
PLC_TASK_6 TASK_INFO_ECLR VAR_GLOBAL Task 6 information %MB1.5640
PLC_TASK_7 TASK_INFO_ECLR VAR_GLOBAL Task 7 information %MB1.5768
PLC_TASK_8 TASK_INFO_ECLR VAR_GLOBAL Task 8 information %MB1.5896
Main:Main
MainV:STD...
Global_Varia...
Formando árbol de instancia para RECURSO 'STD_RES' ...
---------- Generando código específico para la CONFIGURACION STD_CNF ----------
---------- Generando código específico para el RECURSO STD_RES ----------
0 errore(s), 1 aviso(s)
Código Errores Avisos Informaciones Errores PLC Imprimir Multiusuario Configurador de bus FDT
Para obtener ayuda, pulse F1
Inicio
PC WORX - QuickStar...
30.JPG - Paint
ES
10:24

Ilustración 7-3 Resumen de variables de sistema.

Las variables de usuario pueden ser variables locales o variables globales. Las variables locales sólo se pueden solicitar o manipular en el elemento (POU) donde se han creado.

En el caso de las variables de usuario, hay diferentes tipos de variables en función del tipo de POU creada. Encontrará también un resumen en el siguiente cuadro :

POU	Tipo de variable
Función	VAR
	VAR_INPUT
Bloque de funciones	VAR
	VAR_INPUT
	VAR_OUTPUT
	VAR_IN_OUT
	VAR_EXTERNAL
	VAR_EXTERNAL_PG
	VAR_EXTERNAL_FB
Programa	VAR
	VAR_EXTERNAL
	VAR_EXTERNAL_PG

Las variables de usuario, que sean globales, se pueden organizar en varios grupos, para tener así un buen orden de todas ellas. A este nivel, se pueden crear nuevos grupos en el elemento “Global Variables” .

Pulse en la línea “System Variables”, botón derecho del ratón y seleccione el elemento “Insertar grupo”. Aparece una nueva línea con el nombre NewGroup. Cambiad el nombre, haciendo doble clic, poniéndole otro con sentido, por ejemplo, variables del proyecto.

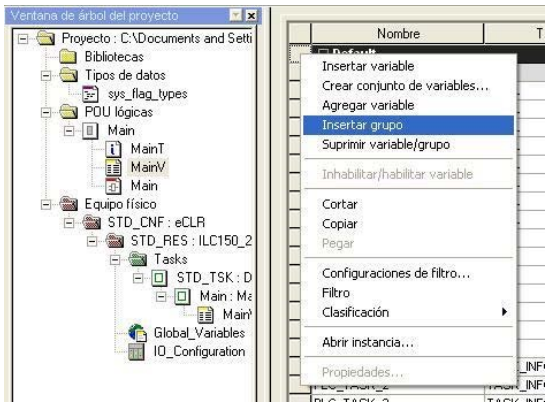


Ilustración 7-4

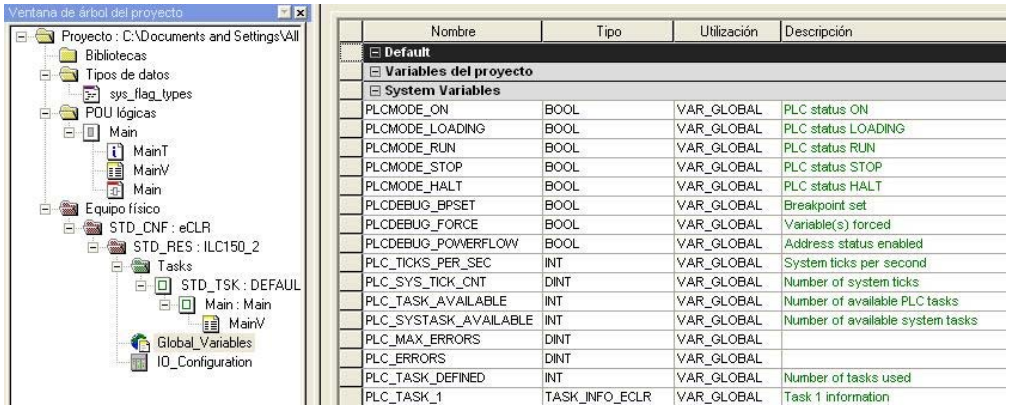


Ilustración 7-5

7.3.3 Creación de un bloque de funciones « Parpadeo »

- La función de parpadeo se crea en un bloque de funciones. Para crear un nuevo bloque de función, seleccione …… ….en el ángulo superior izquierdo.
- Seleccione el lenguaje de programación FBD (“Function Block Diagram” o diagrama de bloques de funciones) e introduzca debajo el nombre del bloque deseado (aquí : ‘Parpadeo’).

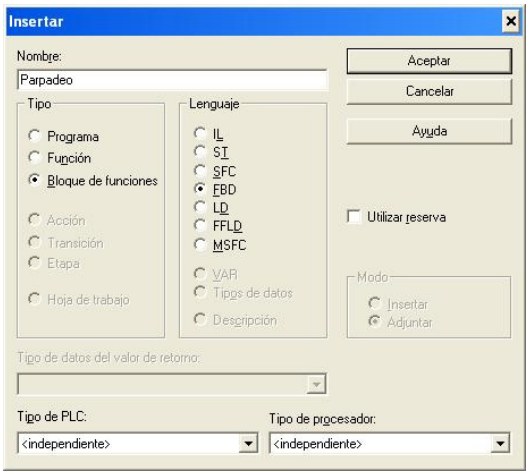


Ilustración 7-6

- Confirme con Aceptar. Se ha creado el bloque.
- Abra la hoja de trabajo del bloque de función clicando dos veces en
- Ahora, se puede escribir el programa para la función de parpadeo.
- Introduzca un temporizador con ‘Retardo a la conexión’ desde el “Asistente de edición”: seleccione el grupo “Bloque de funciones” y pulse dos veces ‘TON’, controle y/o modifice la Instancia del bloque (cada vez que se usa un bloque de funciones, su nombre debe ser diferente) y confirme después con Aceptar.

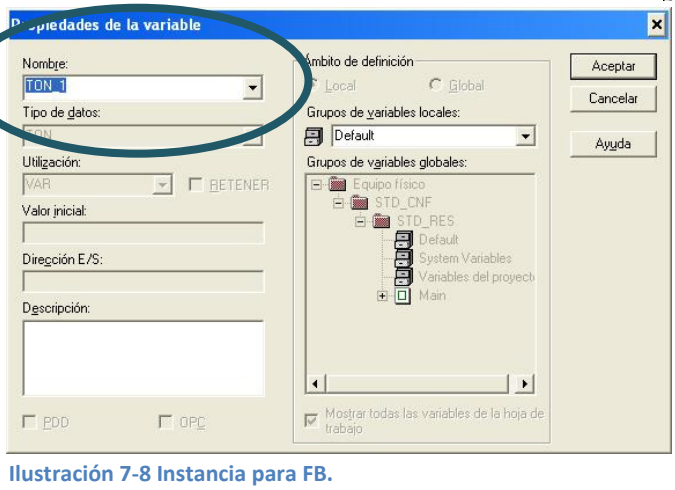
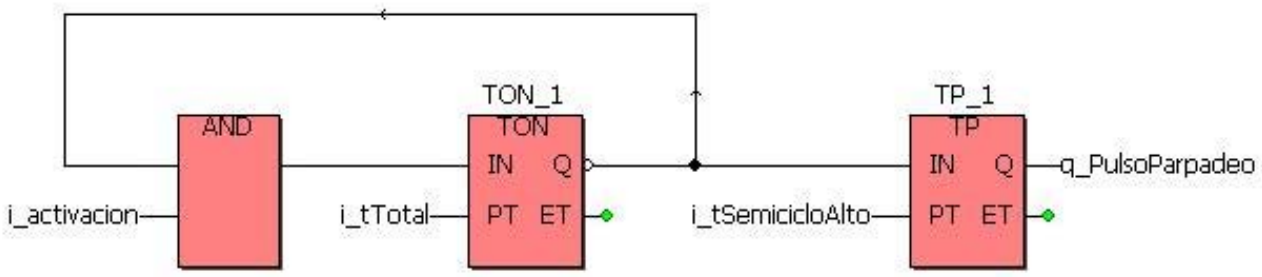


Ilustración 7-8 Instancia para FB.

- Se insertan de la misma forma un temporizador de 'Pulso' y una función AND a partir del Asistente de edición. No olvide que, para la función AND, no hace falta atribuir instancia (diferencia entre el bloque de funciones y la función).
- Finalmente, se conectan los distintos bloques hasta obtener el siguiente sistema (nótese que la salida del TON_1 está negada) :



- Ahora, sólo hay que crear las variables.
- Clique en un lugar de la hoja de trabajo, pulse el botón derecho del ratón y seleccione el elemento "Variable".
- Introduzca el nombre deseado en el ángulo superior izquierdo, por ejemplo : i_activacion.
- Elija como tipo de datos Bool y escoja, del elemento "Utilizacion", VAR_INPUT.

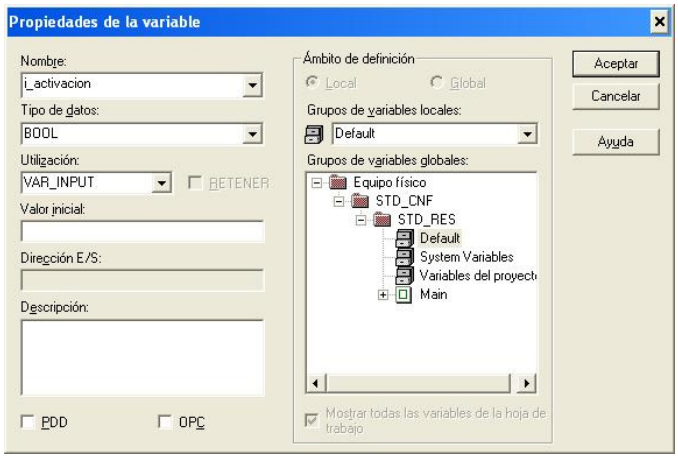


Ilustración 7-9

- Las siguientes variables se declaran de la misma forma :

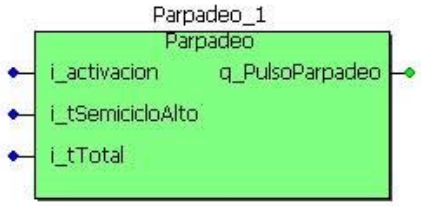
Nombre	Tipo de dato	Utilización	Glob. Var. Grupos
i_activacion	Bool	VAR_INPUT	
i_tTotal	Time	VAR_INPUT	
i_tSemicicloAlto	Time	VAR_INPUT	
q_PulsoParpadeo	Bool	VAR_OUTPUT	

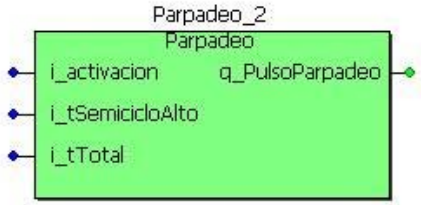
- En ST (Texto Estruturado), este bloque de funciones se parece a :

```
1    TON_1(IN:= NOT TON_1.Q,PT:= i_tTotal);
2
3    TP_1(IN:= TON_1.Q,PT:= i_tSemicicloAlto);
4
5    IF  i_activacion
6        THEN
7            q_PulsoParpadeo := TP_1.Q;
8        ELSE
9            q_PulsoParpadeo := TP_1.Q;
10   END_IF;
```

7.3.4 El programa principal

- Se puede ahora instanciar el bloque de funciones creado en la hoja de trabajo “Main”. Insertela desde el “Asistente de edición” pulsando dos veces en el bloque de funciones ‘Parpadeo’. Este bloque de funciones se puede solicitar, por ejemplo, una segunda vez para hacer parpadear dos salidas. Y así sucesivamente.





- Ahora, aún hay que asignar todas las variables. Las 8 entradas digitales y las 4 salidas digitales en el ILC 150 ETH se encuentran como variables de sistema. **Pulse dos veces en el punto de conexión** (no en el propio nombre) de la entrada i_activacion :

Ilustración 7-10

- Mediante la tecla de selección Nombre, puede escoger entre todas las variables de sistema. Seleccione ONBOARD_INPUT_BIT0.

- Para las variables de entrada del tipo de datos **Time**, se puede, por ejemplo, indicar un tiempo constante.

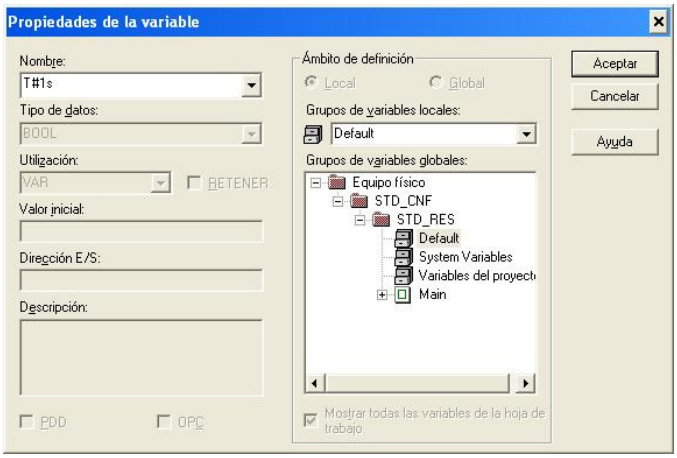
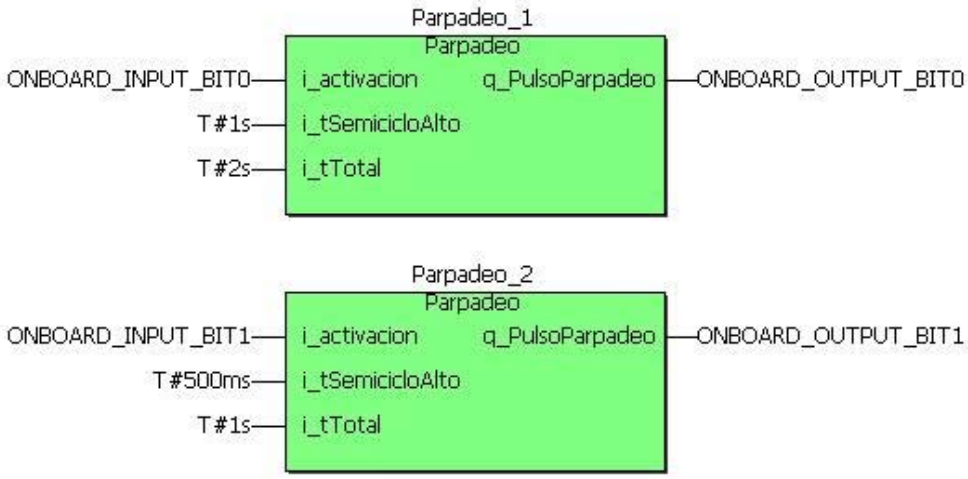


Ilustración 7-11

- El programa se puede finalizar ahora de la siguiente forma :



- Ahora debe compilarse el programa, descargarse al PLC, y realizar un arranque en templado del mismo, tal y como se indicó en el punto 6.8

7.3.5 Pruebas del programa

Finalmente, el programa escrito se puede comprobar. Para ello, active de nuevoel área de trabajo "Programación IEC". Pulsando sobre el icono (Depuración activa/desactiva), puede buscar los errores. En cada variable, se ve ahora su estado y su valor actual.

Activando el primer interruptor de la caja de conmutadores del kit, se puede activar el parpadeo. Se pueden ver ahora las modificaciones de estado, tanto en línea como en los propios módulos.

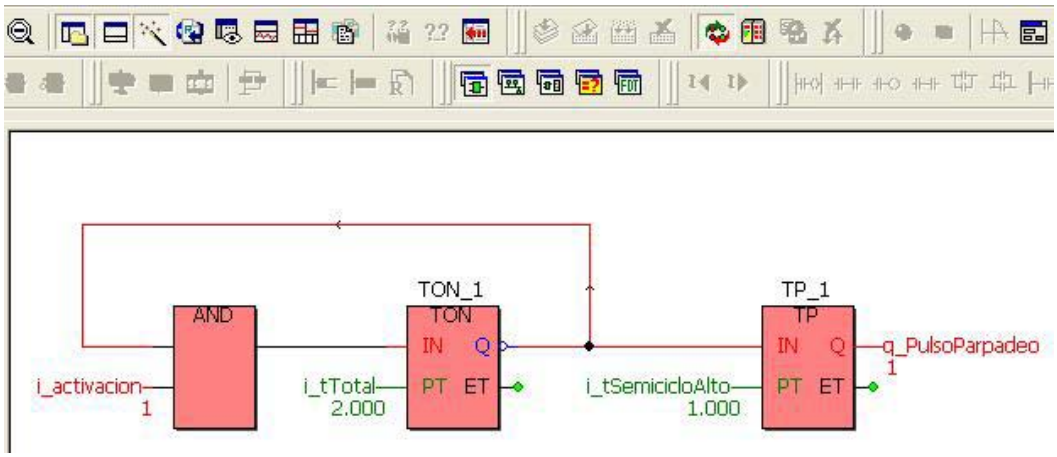
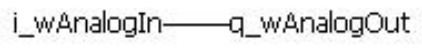


Ilustración 7-12

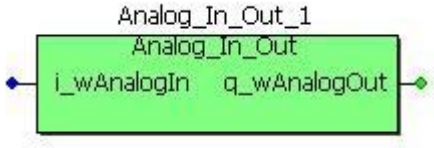
7.3.6 Bloque de funciones para un tratamiento de la señal analógica

Para explicar la Asignación de datos de proceso y la utilización de entradas y salidas analógicas, también prevemos una segunda funcionalidad, un bloque de funciones Analog_In_Out. Para realizarlo, se ha creado un nuevo bloque de funciones con el nombre 'Analog_In_Out'. El objetivo es asociar la entrada analógica (el potenciómetro) a la salida analógica (gráfico de barras LED).

- Este programa puede ser muy sencillo :



- Llamada de este bloque de funciones en el programa 'Main':



- Ahora, hay que crear dos variables para asociarlas a la entrada–salida física real. Éstas se declaran de la forma siguiente y se sitúan en Variables de Proyecto.

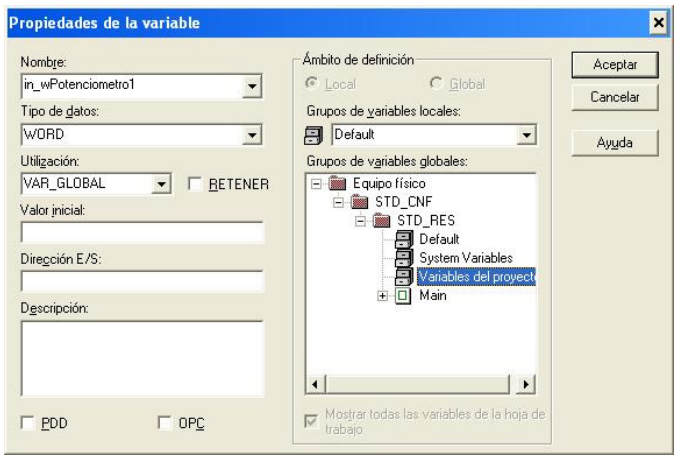
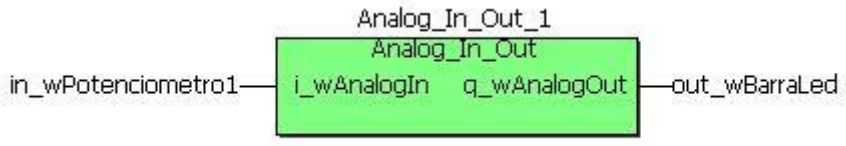


Ilustración 7-13

- Finalmente, se obtiene el siguiente programa :



www.infoPLC.net

Es necesario ahora asignar estas variables a las entradas y salidas físicas del bus. Para ello se procede como se indica en el punto 7.3.7

Ahora, se puede comprobar este programa : Tras los tests, se verifica que hay un problema. Sólo se utiliza la mitad de la barra de LED.

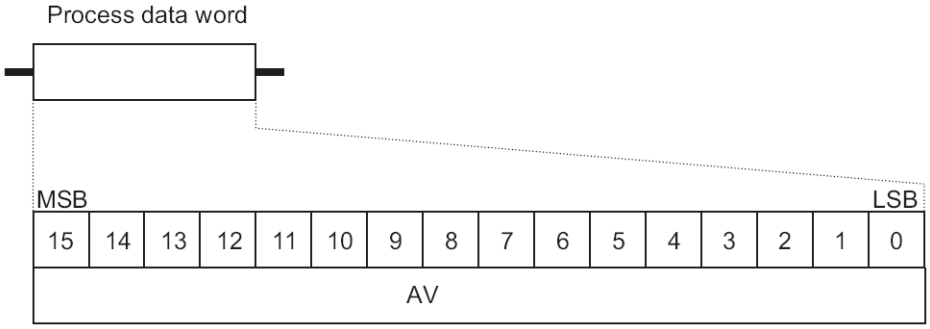
El problema se produce a nivel de los módulos.

Los módulos de entradas analógicas utilizan un bit de signo:

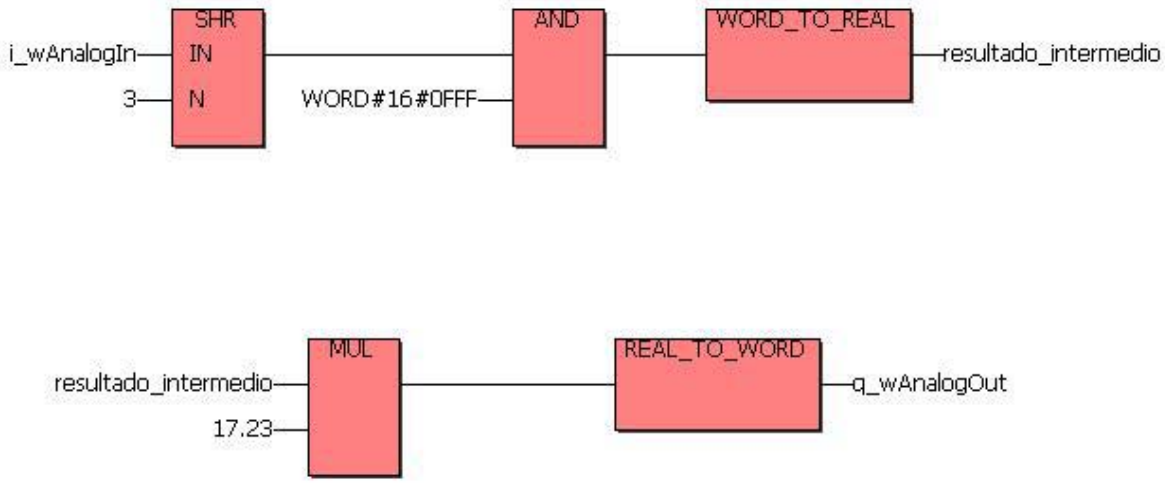
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SB	AV												x	x	x

SB Sign bit AV Analog value
x Without meaning (may be 0 or 1)

El módulo de salida analógica no utiliza un bit de signo:



Por eso, el bloque de funciones se ha adaptado de la siguiente manera :



La misión que se acaba de efectuar es una simple manipulación de datos. Para realizarla, también se puede utilizar una función en vez de un bloque de funciones.

Añada una función, seleccione un nombre y un lenguaje de programación. Fijese en que el nombre de la función sea el mismo que se aplicará a la variable de resultado del bloque, y en que se debe atribuir también un tipo de datos.

Atribución de un tipo de datos al nombre de la función

Ilustración 7-14

La función en lenguaje ST se convierte en el siguiente código :

```
resultado_intermedio := WORD_TO_REAL(SHR(i_wAnalogIn,3) & WORD#16#0FFF);
analogInOut := REAL_TO_WORD(resultado_intermedio * 17.23);
```

7.3.7 Asignación de las variables de proceso

La última etapa de la programación es la asignación de las variables creadas a las entradas y salidas reales (del bus) que forman parte del proyecto. Para realizarla, se pasa al área de trabajo “Asignación de datos de proceso” mediante . Seleccionando STD_RES:…, se muestran todas las variables globales. A partir de la configuración física (reproducida a la derecha), se pueden seleccionar los módulos analógicos y se pueden llamar así las listas de parámetros (ángulo inferior derecho). Clicando en el parámetro deseado y arrastrándolo hacia la izquierda (hacia la variable deseada), se puede realizar la asignación. Para esta aplicación, se deben haber realizado dos asiganciones como las siguientes :

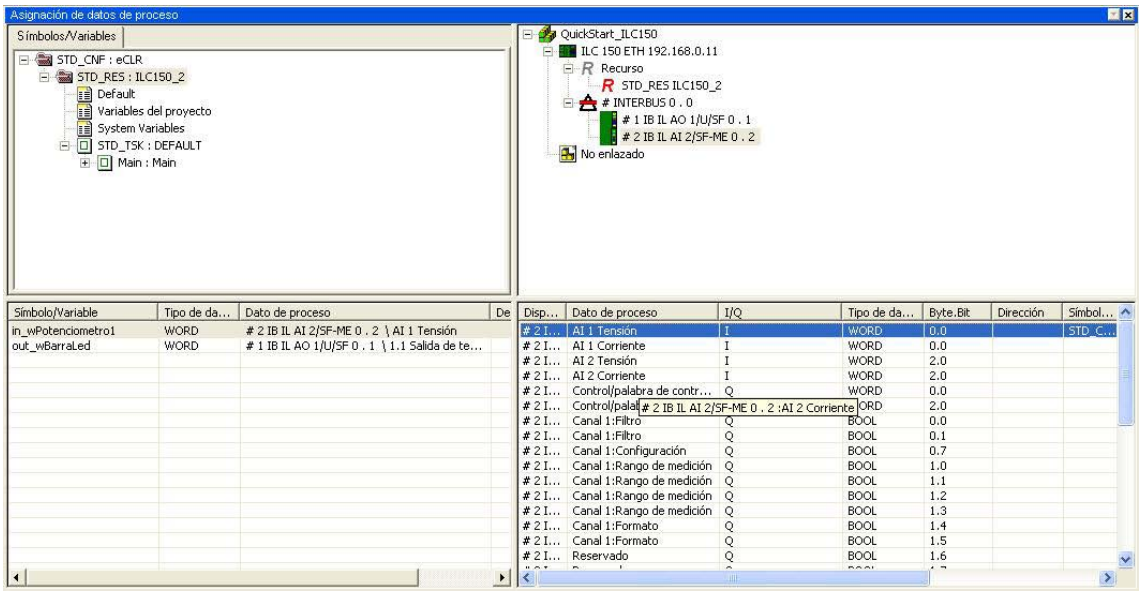


Ilustración 7-15

El programa se puede ahora compilar y descargar al PLC.

7.3.8 Ventana de inspecciones

Otra manera práctica de controlar el programa es la Ventana de inspecciones.

Estando en línea (online), se puede añadir cada variable a la Ventana de inspecciones.

Clique, en el entorno de programación, sobre una variable, pulse con el botón derecho del ratón y seleccione el elemento "Agregar a ventana de inspecciones".

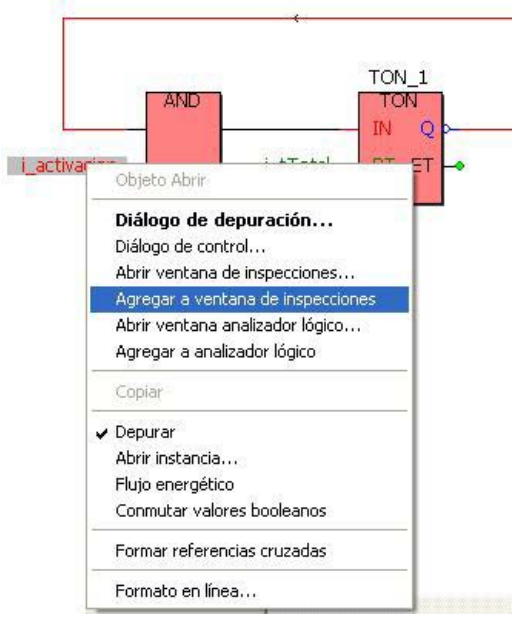


Ilustración 7-16

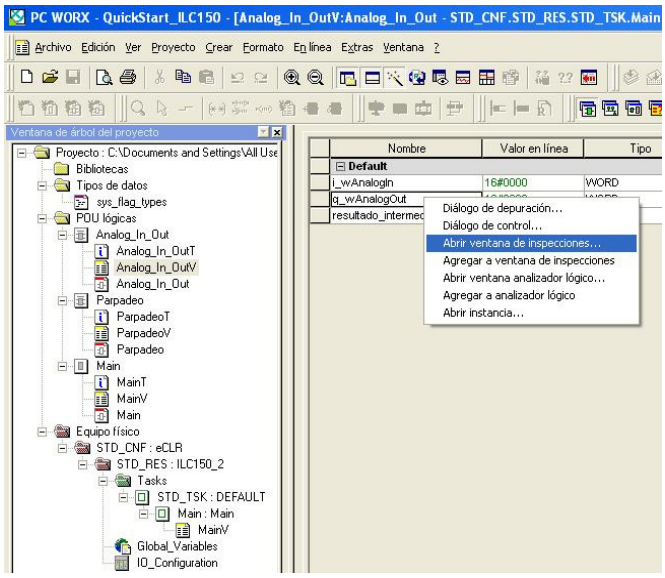


Ilustración 7-17

Otra manera consiste en seleccionar la hoja de trabajo variables de un bloque de funciones, de una función o de un programa determinado, y clicar con el botón derecho del ratón sobre una variable, y elegir el elemento “Agregar a ventana de inspecciones”.

Active la ventana de inspecciones mediante el siguiente icono .

Illustration 7-18

8 Webvisit

8.1 Introducción

Webvisit es un paquete software para generar páginas web. Se puede utilizar aquí para visualizar las variables globales del Controlador Inline creadas con PC WorX.

Una variable, para ser utilizada en la página web, se debe declarar como una VAR_GLOBAL. Además, se ha de seleccionar la "checkbox PPD", ver Ilustración 8-1:

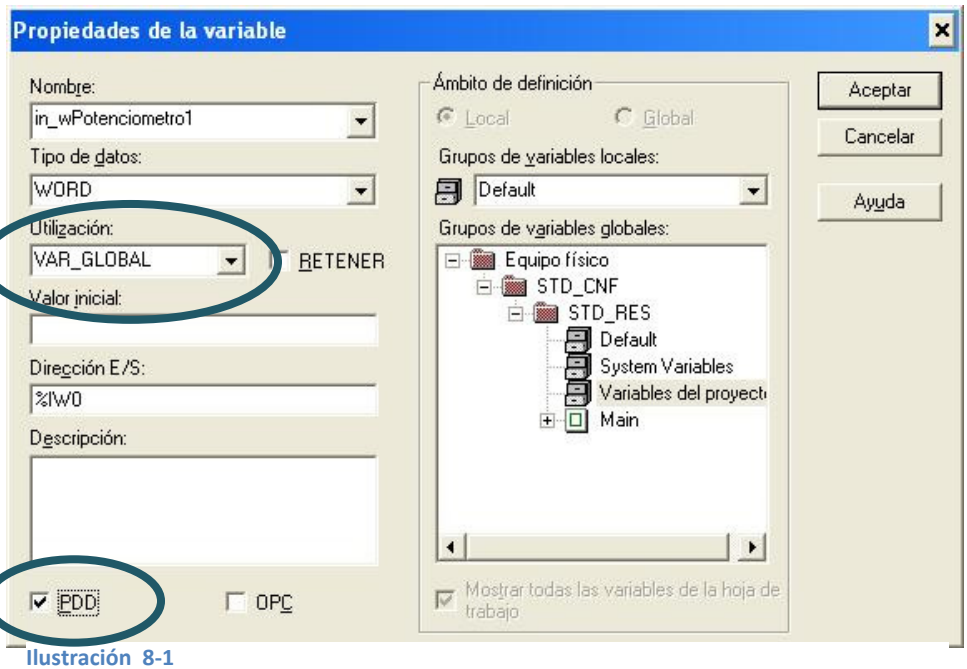


Ilustración 8-1

Se encuentra también un listado de todas variables globales, entre ellas todas las entradas y salidas, en la carpeta “Global_Variables” y ahí se pueden también marcar como PDD.

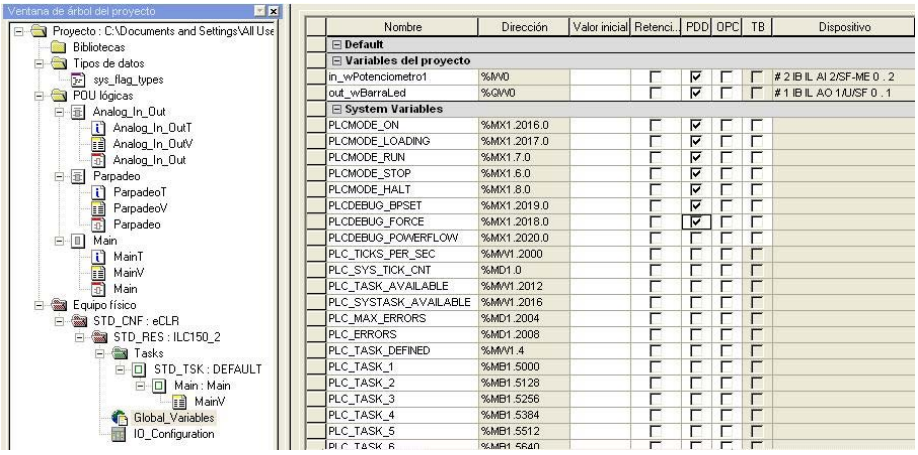


Ilustración 8-2

Para poder activar el bloque de funciones Parpadeo, desde la página Web, cree una variable Global (IN_ArranqueWeb) de tipo booleana, asígnesela a la entrada i_activacion, del bloque de funciones, compile y descargue el proyecto.

Cuando se compila el proyecto, se crea un fichero “pdd.csv”.

La Ilustración 8-3 presenta un ejemplo de un fichero Excel de este género.

Encontrará este fichero en la estructura del proyecto PC WorX. El proyecto PC WorX se encuentra de manera estándar en :

C:\Documents and Settings\All Users\Documentos\PC WORX\

El fichero "pdd.csv” correspondiente se encuentra a continuación en :

C:\Documents and Settings\All Users\Documentos\PC WORX\Projects\xxx\C\STD_CNF\R\STD_RES

con el nombre de proyecto xxx.

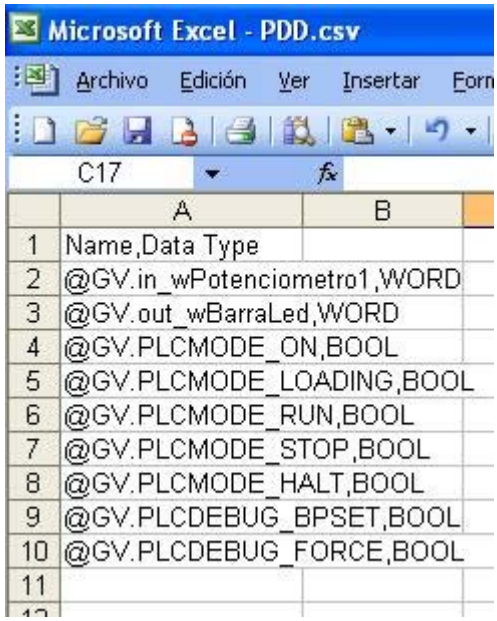


Ilustración 8-3

8.2 Inicio de la visualización

Existe un proyecto WebVisit por defecto en cualquier ILC. Introduciendo la dirección IP del controlador ILC en un navegador de Internet (p.e. Internet Explorer, Mozilla, etc), se puede iniciar la visualización (por ejemplo escriba en la barra de direcciones: “192.168.0.11”).

8.3 Añadir una modificación en WebVisit

Iniciar el WebVisit mediante “Inicio… Todos los programas… Phoenix Contact… AUTOMATIONWORX Software suite 2008 1.40… WebVisit”. El software WebVisit está disponible en inglés.

8.3.1 Crear un nuevo proyecto

- Cierre los proyectos abiertos, si los hay, seleccionando, en “File”, “Close project”. La creación de un nuevo proyecto se realiza seleccionando, en “File”, “New Project…”.
- Seleccione el elemento “Project configurations” en el menú “Project”.

Es importante indicar con qué proyecto PC WorX existirá una comunicación, y cuál es el fichero “PDD.CSV” que incorpora las variables que se utilizarán.

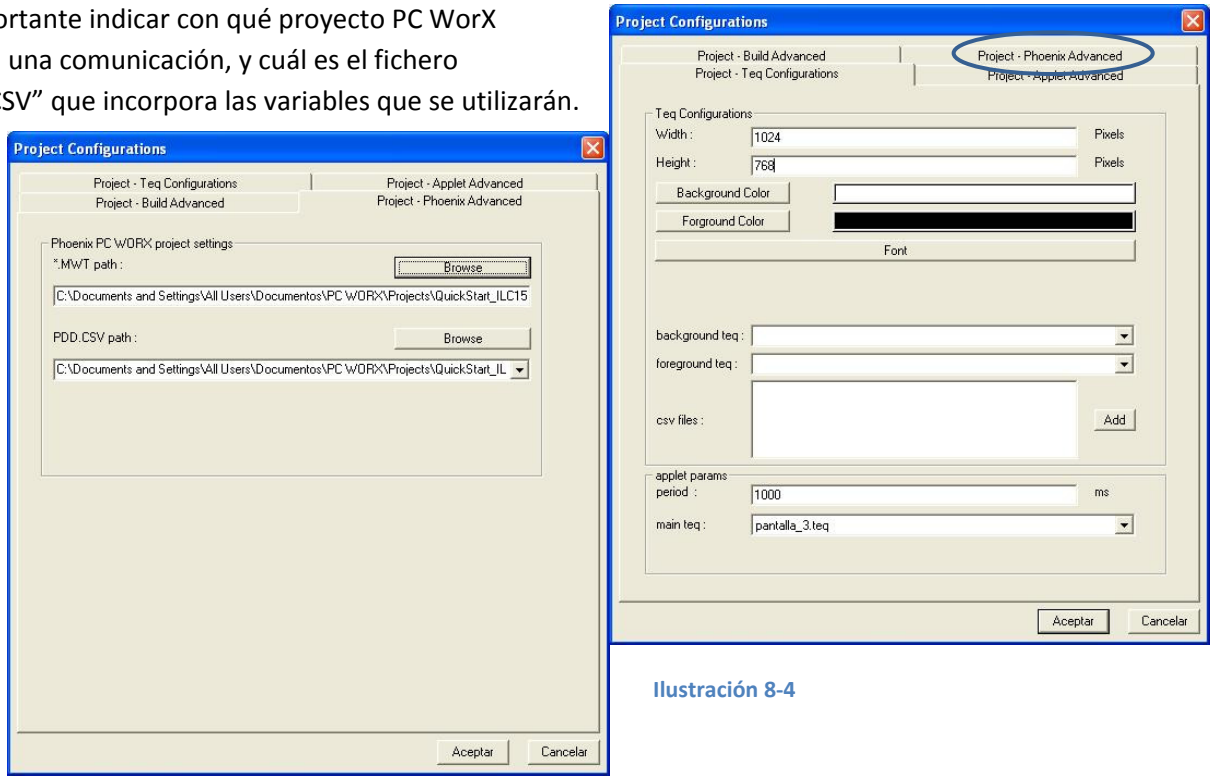


Ilustración 8-4

El proyecto PC WorX :

Ilustración 8-5

El fichero “PDD.CSV” se selecciona automáticamente al elegir el proyecto de PC WorX correspondiente en el paso anterior.

8.3.2 Crear una nueva página web

- La introducción de una nueva página web (xxx.teq) se realiza de la siguiente manera :

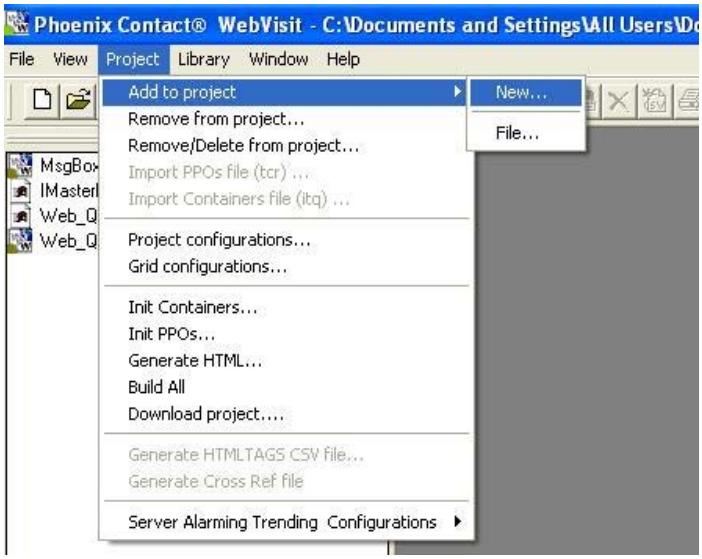


Ilustración 8-6

- A continuación, llame a la página ‘Parpadeo’.
- Seleccione la página que se acaba de crear “Parpadeo.teq”.
- En la ventana, introduzca un botón y un círculo a partir de la barra de herramientas.

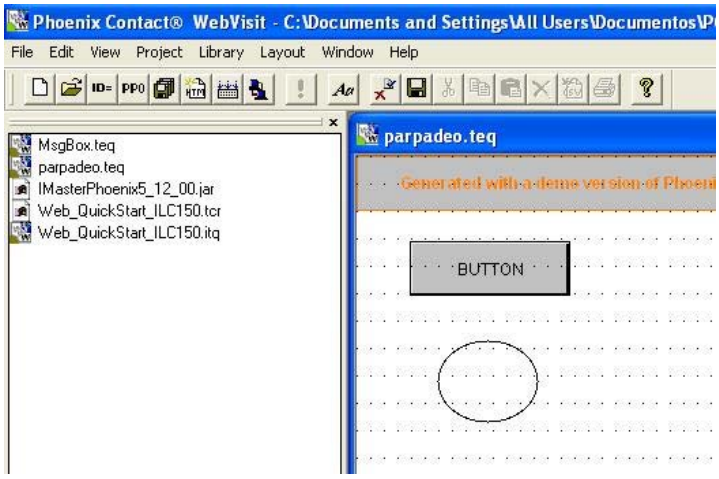


Ilustración 8-7

- Pulse dos veces sobre el botón creado para obtener las propiedades.
- En “Repaints”, se puede modificar el texto del botón.

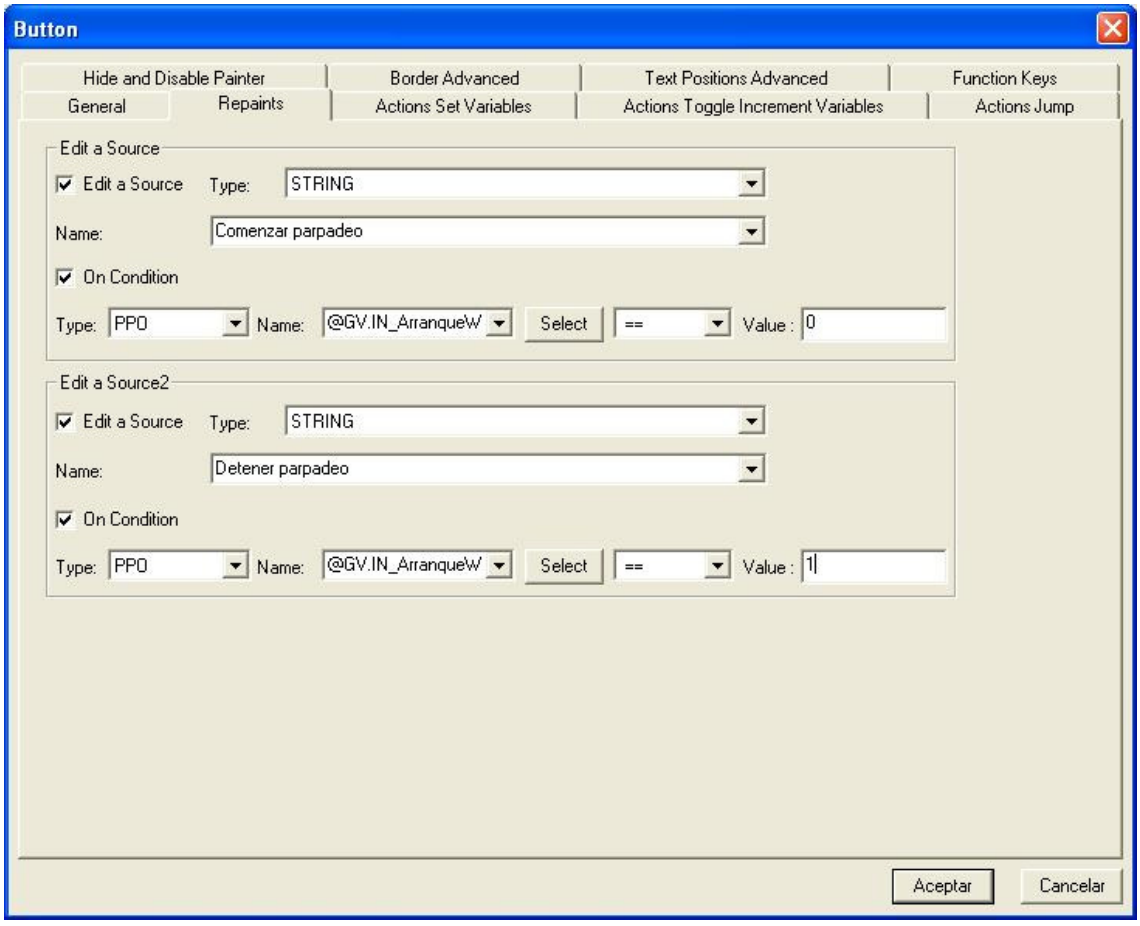


Ilustración 8-8

- Cuando se acciona el botón select, aparece el contenido del fichero “PDD.SCV”. Ahora, la variable necesaria del programa PC WorX se puede asociar a una acción determinada en WebVisit (aquí “IN_ArranqueWeb”).

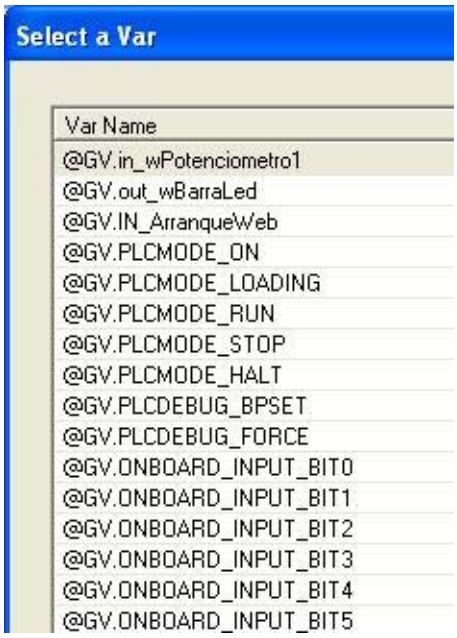


Ilustración 8-9

- Reajuste de la acción a realizar desde el mando del botón (funciona como un conmutador: toggle) :

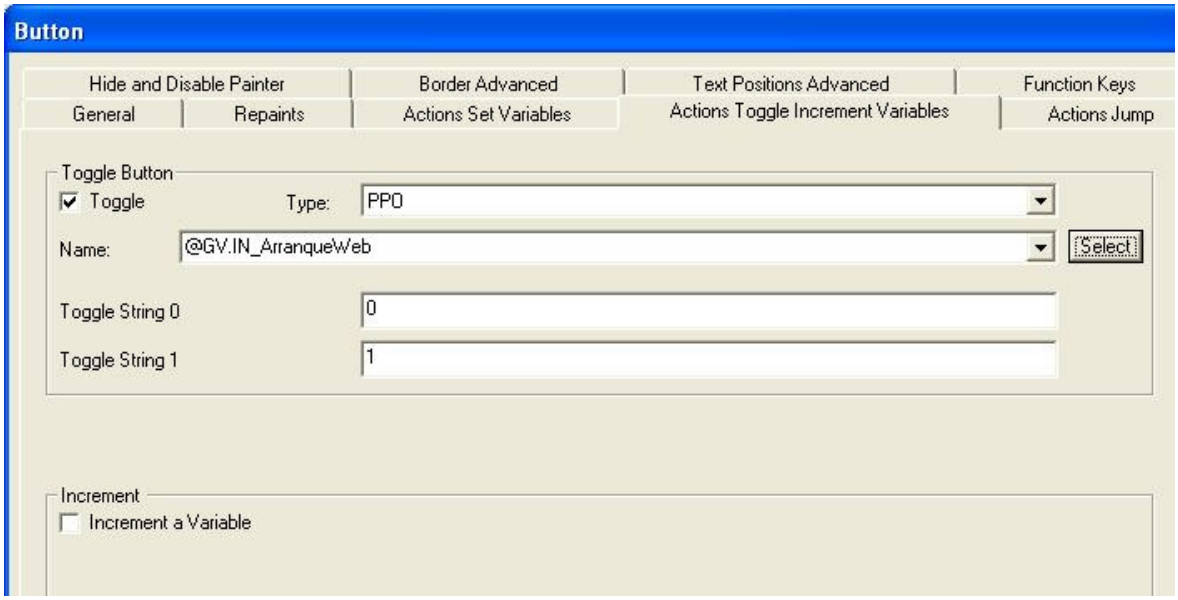


Ilustración 8-10

- Modificar el color del botón si está activado o no.

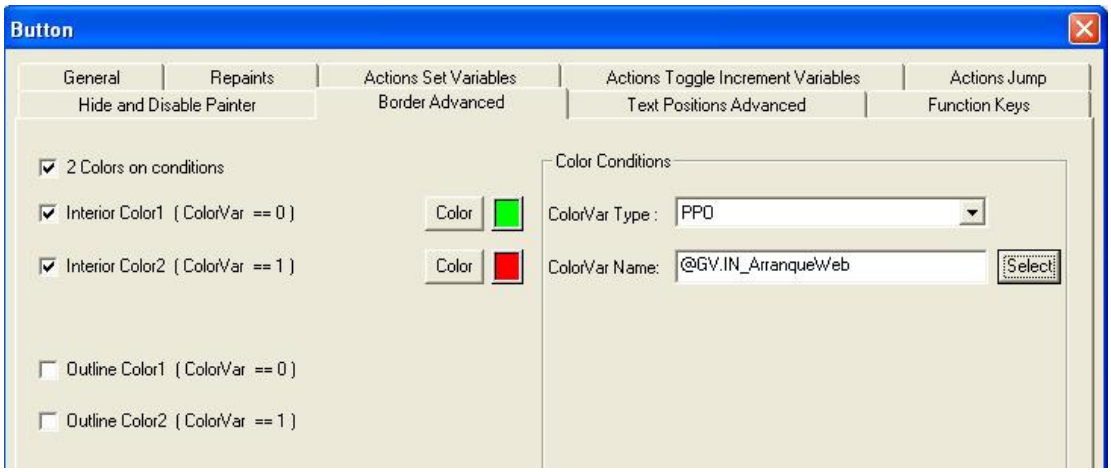


Ilustración 8-11

- Asociar el círculo a la variable “ONBOARD_OUTPUT_BIT0”.

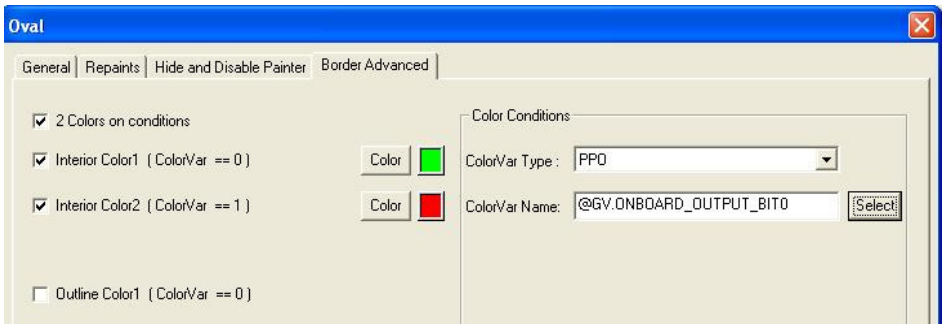


Ilustración 8-12

Finalmente, todavía se debe realizar un ajuste para conseguir que la página que se acaba de crear se reproduzca como página de inicio. Mediante “Project… Project Configurations…”, se llama a la ventana de más abajo. Aquí, el “Main teq” se puede ajustar. Introduzca, para ello, ‘Parpadeo’, para que se convierta en la página de inicio.

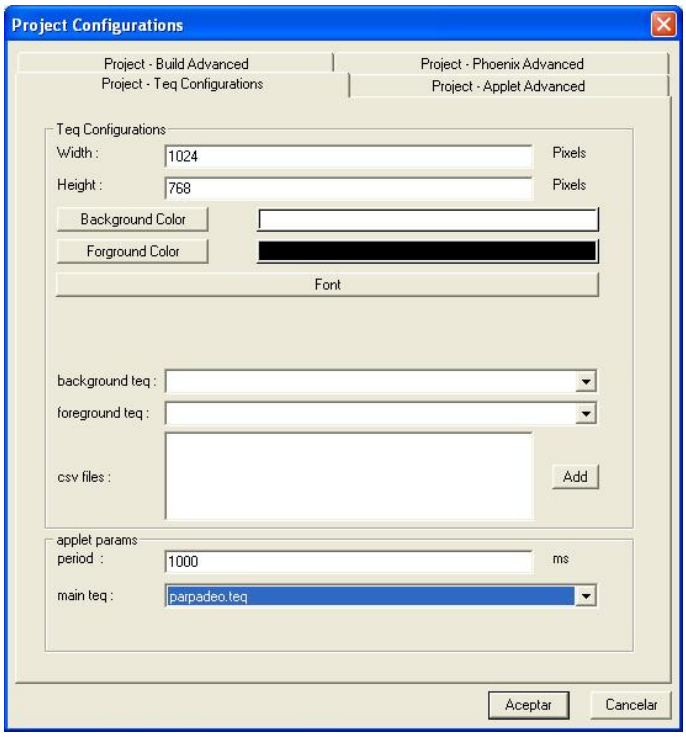


Ilustración 8-13

- Para visualizar el valor de salida analógica, se utilizará un “Bar graph” (histograma). Añadid un histograma en la forma indicada en la ilustración siguiente :

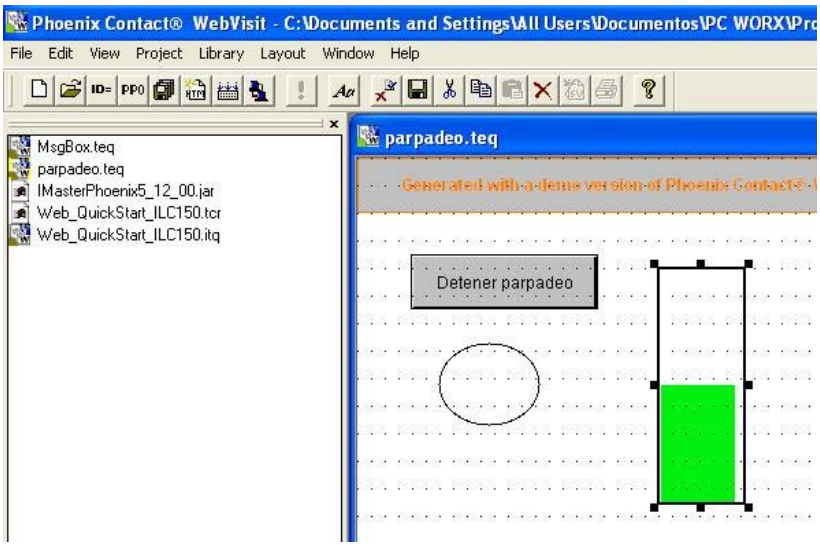


Ilustración 8-14

- La parametrización del diagrama se realiza de nuevo clicando dos veces debajo. En “Repaints”, se puede ajustar la variable adecuada (escogiendo tipo PPO y con el botón Select). El diagrama variará a la vez que la variable PLC ‘out_wBarraLed’ :

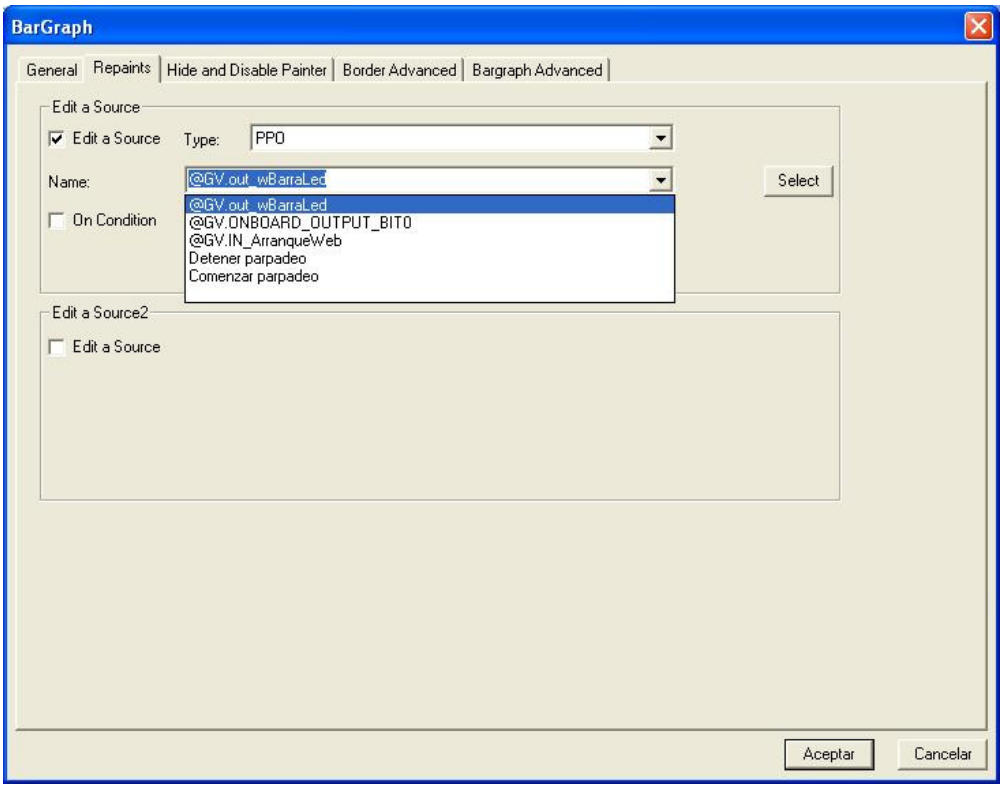


Ilustración 8-15

- En la pestaña “Bargraph Advanced”, se pueden ajustar los límites entre los que variará la variable. Para esta salida, se realizará entre 0 y 64000 :

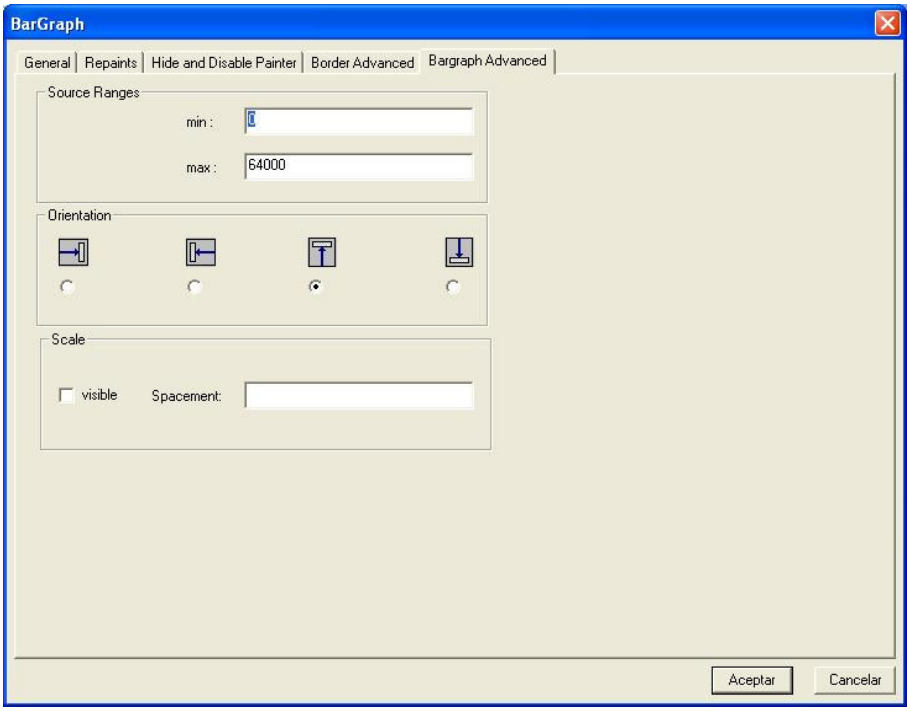


Ilustración 8-16

- La dirección en que debe aumentar el diagrama se puede regular también con el ajuste “Orientation”.

8.3.3 Descarga del proyecto

Para descargar el proyecto en el ILC 150 ETH, hay que proceder de la siguiente forma :

- La primera vez que se descarga el proyecto debe generarse el archivo llamado “entry.html”. Es fundamental mantener dicho nombre, o el proyecto no funcionará. Para generarlo se llama a la ventana siguiente:

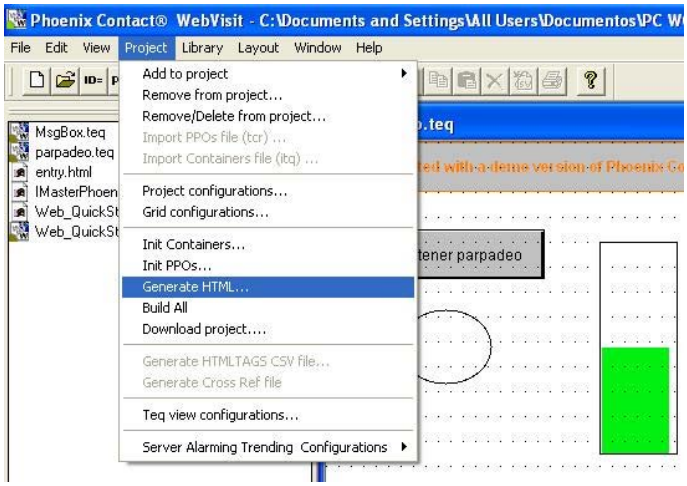


Ilustración 8-18

- Llame a la ventana de descarga :

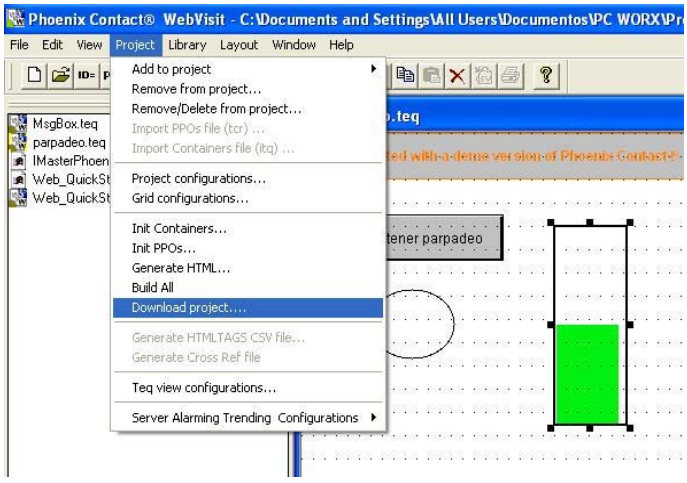


Ilustración 8-19

- Introduzca la dirección IP del ILC (192.168.0.11) y pulse en “Connect”.
- La conexión se ha creado y se carga el proyecto pulsando el botón “Download Project”.

9 Trucos y recursos

9.1 Gestión de la pantalla

- Abreviaturas en el área de trabajo

  <SHIFT> F2 Asistente de edición
  <ALT> F2 Ventana de mensajes
  <CTRL> F2 Referencias cruzadas

  <SHIFT> F8 Ventana de árbol del proyecto
  <ALT> F10 Ventana de inspecciones

- Si las pantallas están totalmente desordenadas, se pueden reorganizar completamente.

  En el menú “Extras” seleccione “Opciones” y elija la pestaña “General”.
  Clique luego “Restaurar área de trabajo”.

  A continuación, hay que volver a iniciar el PC WORX.

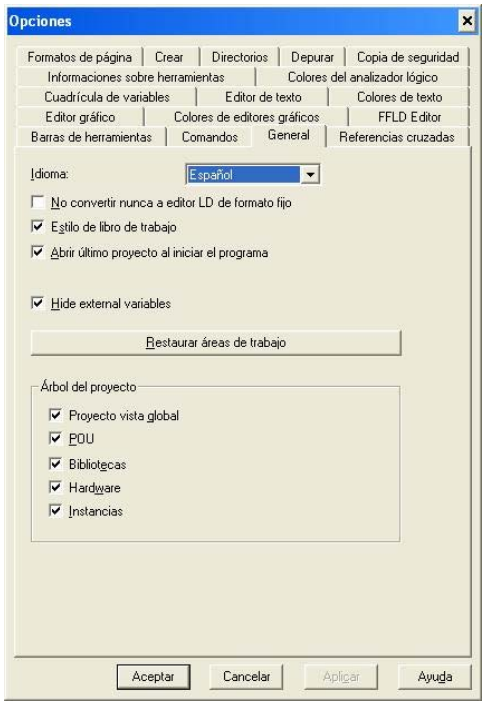


Ilustración 9-1

- En la ventana de programación, es fácil utilizar el zoom pulsando la tecla <CTRL> y usando a la vez el ratón.

9.2 Gestión de las variables

- Supresión de las variables e instancias que no se utilizan. Seleccione “Eliminar variables … “ en “Crear”.

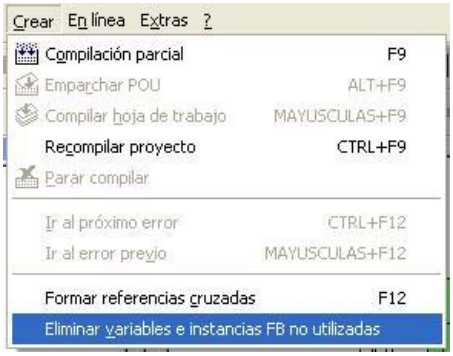
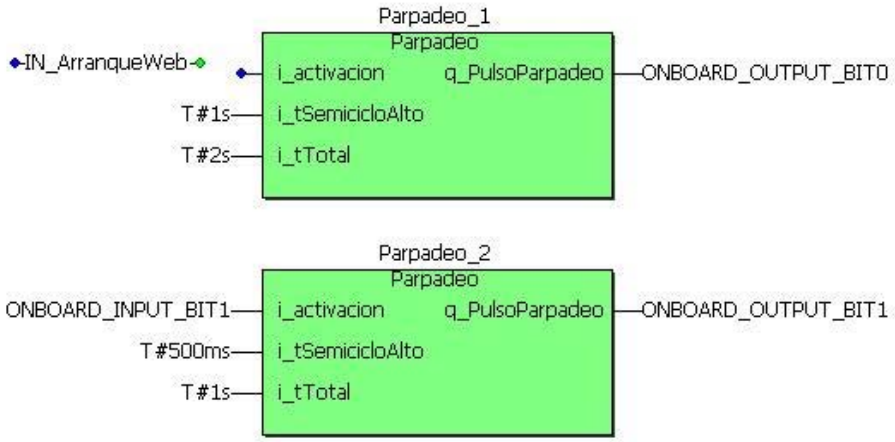

Crear En línea Extras ?
Compilación parcial F9
Emparchar POU ALT+F9
Compilar hoja de trabajo MAYUSCULAS+F9
Recompilar proyecto CTRL+F9
Parar compilar
Ir al próximo error CTRL+F12
Ir al error previo MAYUSCULAS+F12
Formar referencias cruzadas F12
Eliminar variables e instancias FB no utilizadas

Ilustración 9-2

- En FBD

  Para separar las variables, pulse la tecla <SHIFT>, clique en la variable y arrastrela.



  Para copiar una variable, pulse la tecla <CTRL>, clique en la variable y arrastrela.

Para seleccionar un bit, o un byte dentro de una word (palabra) : clique en la bolita de una entrada o salida, se abre la ventana de Propiedades de las variables, seleccione la variable de tipo word, escriba un punto detrás de su nombre y aparecerá una lista de tpdos los bits y bytes que forman parte de esta palabra.

Ilustración 9-3

o En ST

La tecla <CRLT> y la barra espaciadora proporcionan una lista de todas las variables.

Seleccione una variable de tipo word, escriba un punto detrás y aparecerán todos los bits y bytes que se puedan seleccionar en esta palabra.

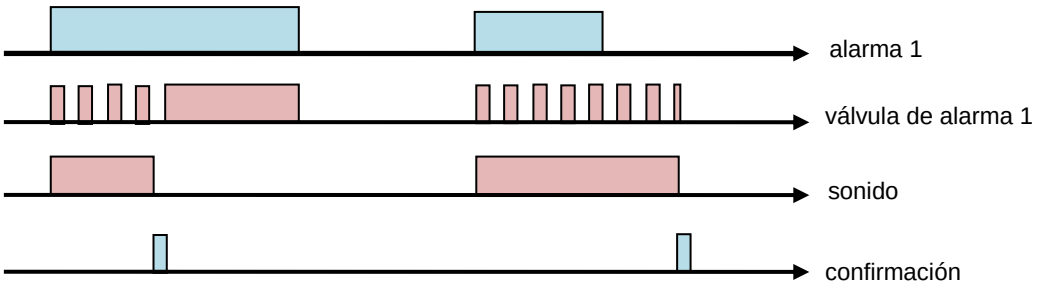
www.infoPLC.net

10 Ejercicios

10.1 Un sistema de alarma

- Un sistema de alarma tiene 4 entradas, 4 luces de alarma, un sonido, una entrada de confirmación y un botón de test.
- Si se activa una de las entradas de alarma :
  - se encenderá una de las luces de alarma, de forma intermitente cada 500ms.
  - se accionará un sonido general.
- Tras la confirmación de alarma :
  - la luz correspondiente se enciende de forma continua
  - el sonido cesa
- Tras la desaparición de la causa de la alarma, se apaga la luz de alarma correspondiente.
- Si desaparece una alarma antes de que el operador haya podido confirmarla, el sonido y la luz continúan activos.
- Un botón de test permite realizar un test de la luz y un test sonoro. Cuando se activa este botón, se encienden todas las luces. Cuando se suelta este botón, las luces se vuelven a apagar.

El programa final (en cualquier caso) debe satisfacer el siguiente diagrama temporal :



- El siguiente diagrama se ha elaborado para una alarma. Escriba un bloque de funciones para ejecutar lo que se ha solicitado.
- Realice un programa para las cuatro luces de alarma.

Utilice la función parpadeo que ya se ha creado en este curso.

Realice a continuación un bloque de funciones para el tratamiento de la alarma. Se encuentra una posible solución para este bloque en la Ilustración 10-1 Solución FB tratamiento de la alarma.

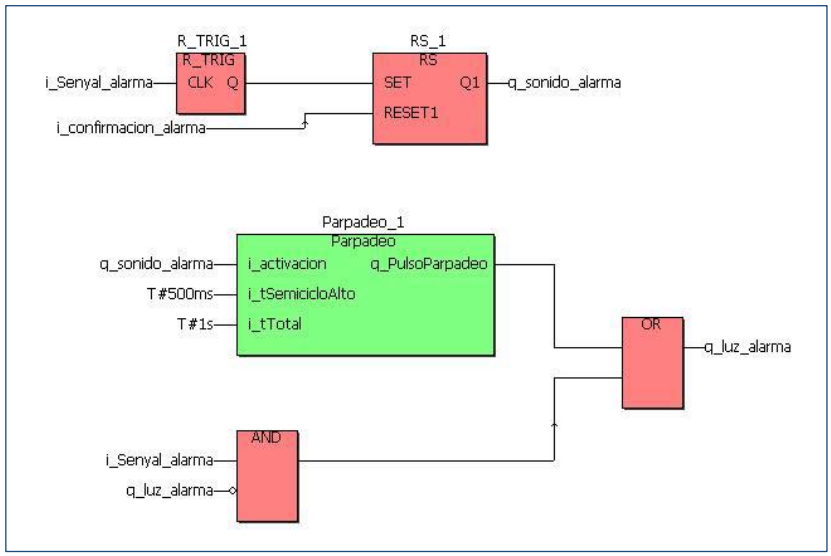


Illustration 10-1 Solución FB tratamiento de la alarma

Todo el programa :

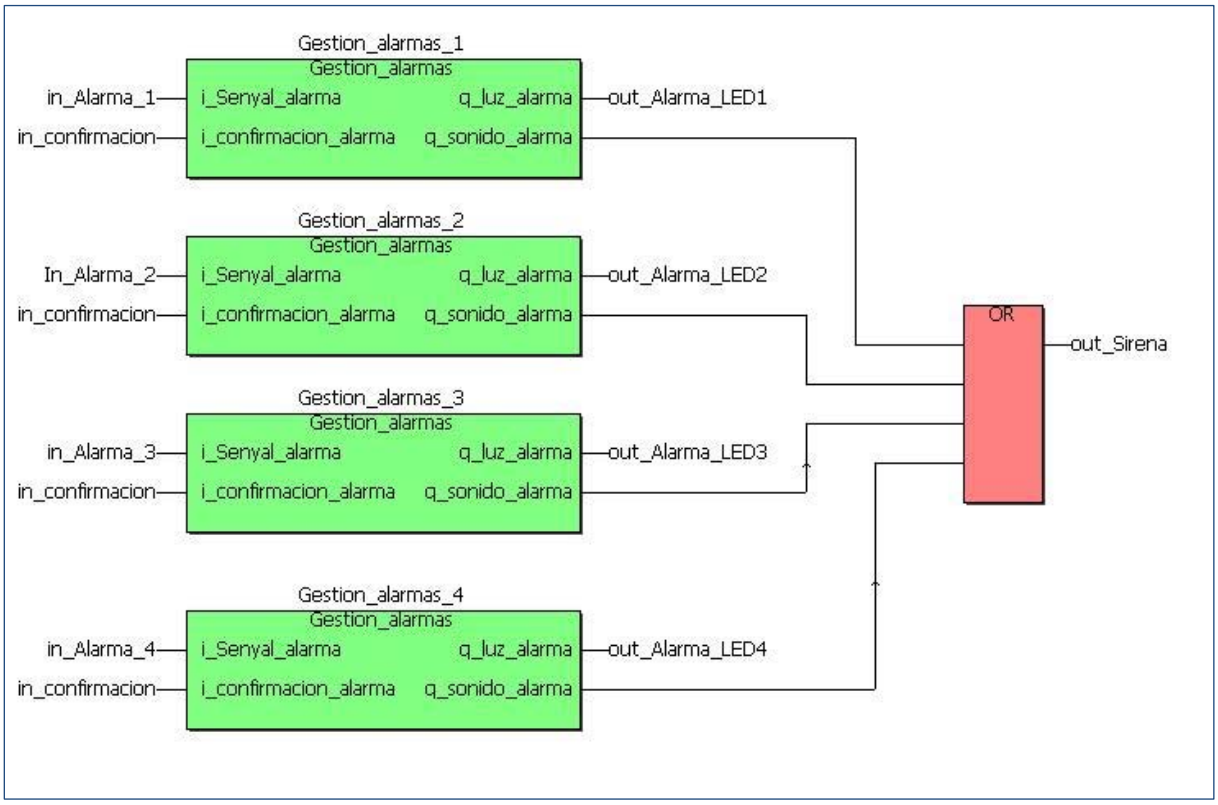


Ilustración 10-2 Solución completa.

10.2 Apertura de un portón

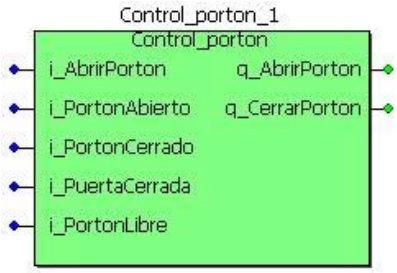
Se debe activar automáticamente un portón. Mediante un botón de arranque, se ha de abrir el portón. Cuando se abra, debe cerrarse automáticamente en 10 segundos.
Se han de controlar las siguientes condiciones :

- Hay una puerta en el portón. Un captador verifica que la puerta esté cerrada. El portón no puede abrirse si la puerta no está cerrada.
- Los contactos de final de carrera señalan la apertura y el cierre del portón.

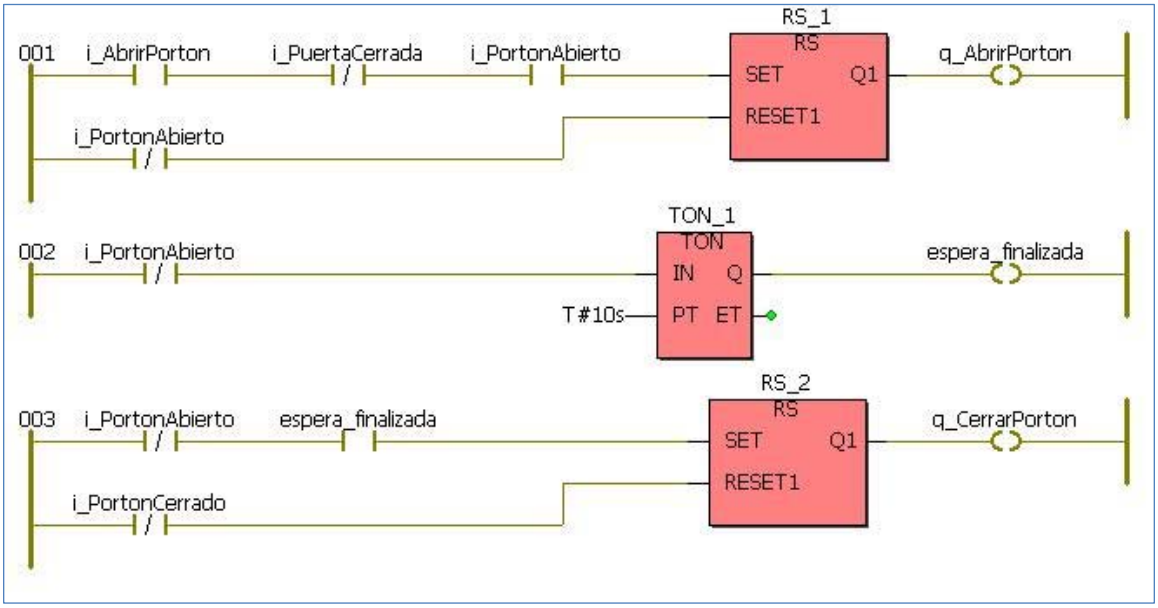
Lista de entradas–salidas :

Función del indicador	I/Q	Tipo	Accionado	Nombre lógico
Botón de apertura del portón	I	NA	Detención '1'	in_AbrirPorton
Final de recorrido de cierre del portón	I	NC	Detención '0'	in_PortonCerrado
Final de recorrido de apertura del portón	I	NC	Detención '0'	in_PortonAbierto
Contacto de la puerta	I	NC	Detención '0'	in_PuertaCerrada
Apertura del portón	Q			out_AbrirPorton
Cierre del portón	Q			out_CerrarPorton

Escribid un bloque de funciones adaptado :

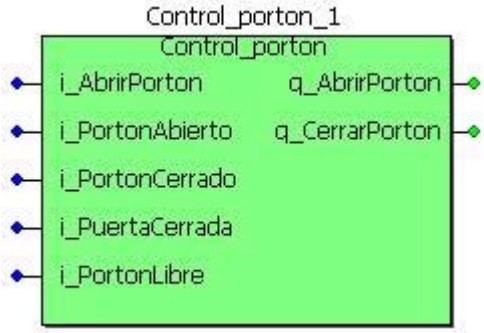


Código del programa del bloque de funciones en LD :



Desarrollo :

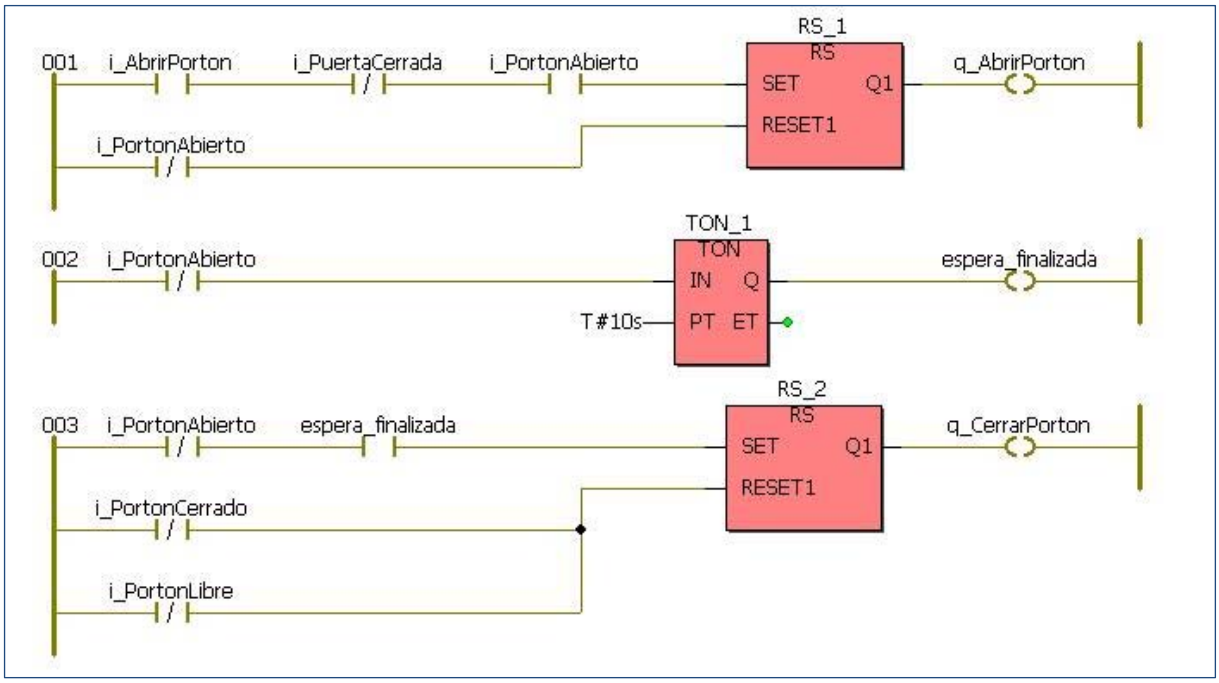
Un captador óptico controla si hay personas u objetos presentes en la apertura del portón. Naturalmente, el portón sólo puede estar cerrado si el paso está completamente libre. Tras la liberación del portón, quedan 10 segundos para que el portón se cierre.



Lista de entradas–salidaa :

Función del indicador	I/Q	Tipo	Accionado	Nombre lógico
Botón de apertura del portón	I	NA	Detención ‘1’	in_AbrirPorton
Final de recorrido de cierre del portón	I	NC	Detención ‘0’	in_PortonCerrado
Final de recorrido de apertura del portón	I	NC	Detención ‘0’	in_PortonAbierto
Contacto de la puerta	I	NC	Detención ‘0’	in_PuertaCerrada
Captador óptico del portón libre	I	NO	Detención ‘1’	in_PortonLibre
Apertura del portón	Q			out_AbrirPorton
Cierre del portón	Q			out_CerrarPorton

Código del programa adaptado del bloque de funciones en LD :



10.3 Control de la temperatura

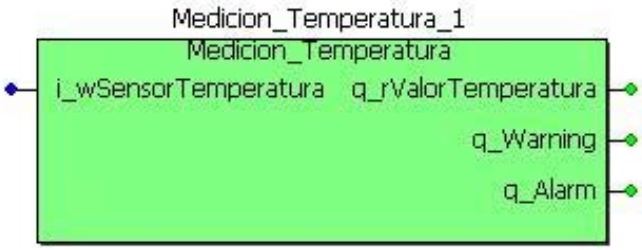
Un captador de temperatura proporciona una señal de corriente que va de 4mA a 20mA (0 .. 30.000) y ello para temperaturas que van de -50 °C a +100 °C. Con este captador, se debe controlar la temperatura del proceso, y se ha de mantener entre -25 °C y +55 °C.

Mediante el servidor web en el ILC150, se puede controlar el valor real de la temperatura y existen dos avisadores de la situación.

Un avisador 'WARNING', si la temperatura se sitúa entre -10 °C y -25 °C o si la temperatura se sitúa entre +40 °C y +55 °C.

Un avisador 'ALARM', si la temperatura es inferior a -25 °C o si la temperatura es superior a 55 °C.

Escribid un bloque de funciones con tres VAR_OUTPUT's, uno para el valor real de la temperatura en °C y dos para los avisadores.



Código del programa del bloque de funciones en ST :

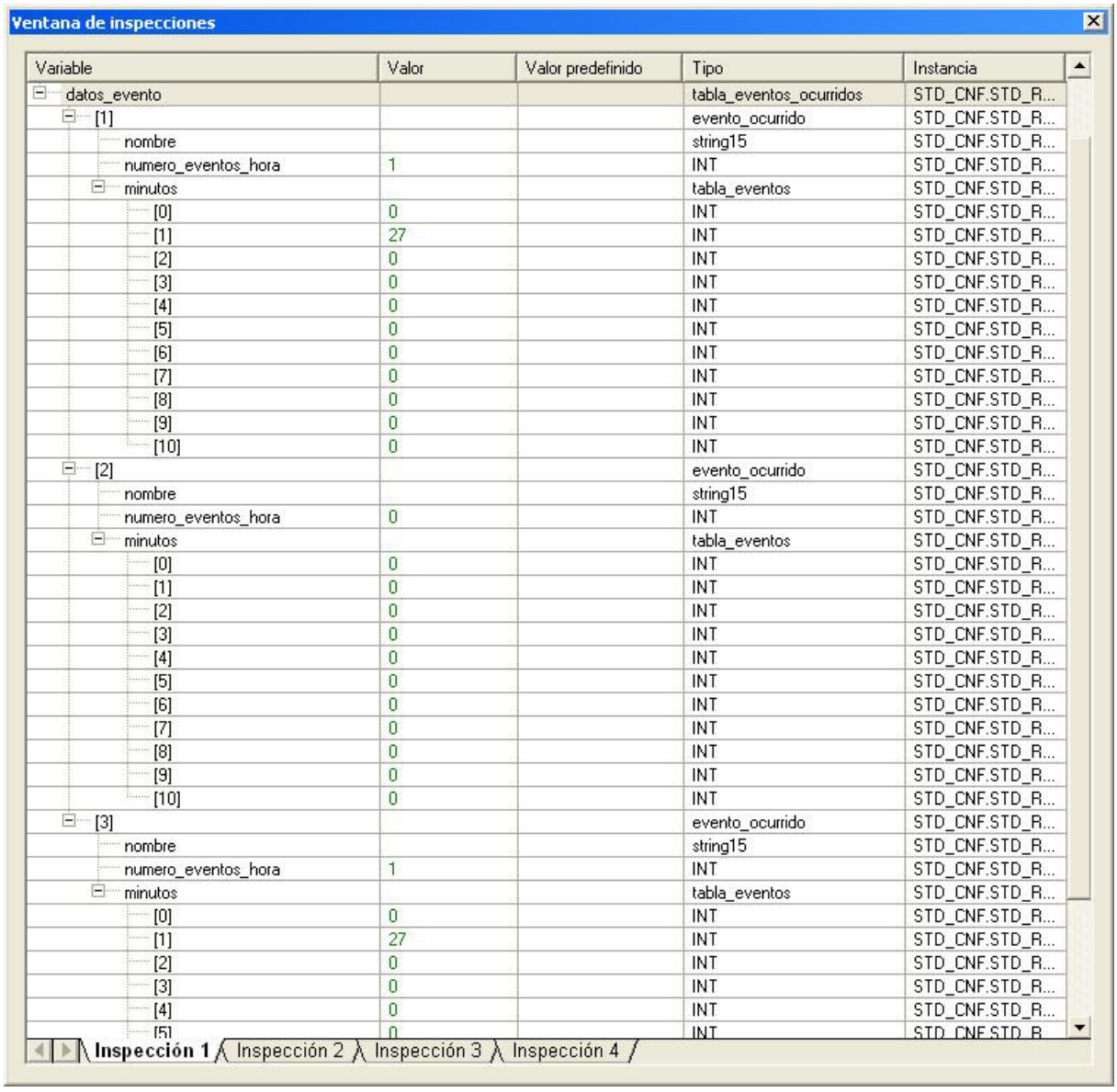
```
(* ajuste del valor del sensor a medicion en °C *)
q_rValorTemperatura := ((WORD_TO_REAL (i_wSensorTemperatura)/30000.0)*150.0)-50.0;

(* condiciones para activar la indicacion WARNING *)
IF  (q_rValorTemperatura < -10.0 AND q_rValorTemperatura > -25.0) OR
    (q_rValorTemperatura > 40.0 AND q_rValorTemperatura < 55.0)
    THEN
        q_Warning := TRUE;
    ELSE
        q_Warning := FALSE;
END_IF;

(* condiciones para activar la indicacion ALARM *)
IF  q_rValorTemperatura < -25.0 OR q_rValorTemperatura > 55.0
    THEN
        q_Alarm := TRUE;
    ELSE
        q_Alarm := FALSE;
END_IF;
```

10.4 Ejemplo en ST con los array y las estructuras

Objetivo : se deben controlar tres eventos en una instalación. Cada hora, se ha de conservar el número de veces en que ha aparecido uno de estos eventos, así como en qué minuto. Estos datos se pueden borrar en cualquier momento.

Para cada evento, se han de conservar el nombre, el número de veces por hora y un cuadro con la hora.

Finalmente, se ha de obtener la siguiente estructura :



Variable	Valor	Valor predefinido	Tipo	Instancia
datos_evento			tabla_eventos_ocurridos	STD_CNF.STD_R...
[1]			evento_ocurrido	STD_CNF.STD_R...
nombre			string15	STD_CNF.STD_R...
numero_eventos_hora	1		INT	STD_CNF.STD_R...
minutos			tabla_eventos	STD_CNF.STD_R...
[0]	0		INT	STD_CNF.STD_R...
[1]	27		INT	STD_CNF.STD_R...
[2]	0		INT	STD_CNF.STD_R...
[3]	0		INT	STD_CNF.STD_R...
[4]	0		INT	STD_CNF.STD_R...
[5]	0		INT	STD_CNF.STD_R...
[6]	0		INT	STD_CNF.STD_R...
[7]	0		INT	STD_CNF.STD_R...
[8]	0		INT	STD_CNF.STD_R...
[9]	0		INT	STD_CNF.STD_R...
[10]	0		INT	STD_CNF.STD_R...
[2]			evento_ocurrido	STD_CNF.STD_R...
nombre			string15	STD_CNF.STD_R...
numero_eventos_hora	0		INT	STD_CNF.STD_R...
minutos			tabla_eventos	STD_CNF.STD_R...
[0]	0		INT	STD_CNF.STD_R...
[1]	0		INT	STD_CNF.STD_R...
[2]	0		INT	STD_CNF.STD_R...
[3]	0		INT	STD_CNF.STD_R...
[4]	0		INT	STD_CNF.STD_R...
[5]	0		INT	STD_CNF.STD_R...
[6]	0		INT	STD_CNF.STD_R...
[7]	0		INT	STD_CNF.STD_R...
[8]	0		INT	STD_CNF.STD_R...
[9]	0		INT	STD_CNF.STD_R...
[10]	0		INT	STD_CNF.STD_R...
[3]			evento_ocurrido	STD_CNF.STD_R...
nombre			string15	STD_CNF.STD_R...
numero_eventos_hora	1		INT	STD_CNF.STD_R...
minutos			tabla_eventos	STD_CNF.STD_R...
[0]	0		INT	STD_CNF.STD_R...
[1]	27		INT	STD_CNF.STD_R...
[2]	0		INT	STD_CNF.STD_R...
[3]	0		INT	STD_CNF.STD_R...
[4]	0		INT	STD_CNF.STD_R...
[5]	0		INT	STD_CNF.STD_R...

Para realizarla, se han de crear los siguientes tipos de datos :

```
TYPE
        tabla_eventos: ARRAY [0..10] of int;
        string15 : string(15);
        evento_ocurrido:
                STRUCT
                        nombre : string15;
                        numero_eventos_hora : INT;
                        minutos : tabla_eventos;
                END_STRUCT;
        tabla_eventos_ocurridos:        ARRAY [1..3] of evento_ocurrido;
END_TYPE
```

A continuación, se crea un programa.

Se crea una variable con el nombre 'datos_evento' del tipo de datos 'tabla_eventos_ocurridos'.

Solución en ST :

```
if  hora_ciclo_anterior <>  RTC_HOURS  then
        for num_evento := 1 to 3 do
                datos_evento[num_evento].numero_eventos_hora := 0;
                for index := 1 to 10 do
                        datos_evento[num_evento].minutos[index] := 0;
                end_for;
        end_for;
end_if;
```

```
hora_ciclo_anterior:= RTC_HOURS;


R_TRIG_1(CLK:= ONBOARD_INPUT_BIT0);
R_TRIG_2(CLK:=  ONBOARD_INPUT_BIT1);
R_TRIG_3(CLK:= ONBOARD_INPUT_BIT2);


if R_TRIG_1.Q then
        index := datos_evento[1].numero_eventos_hora +1;
        datos_evento[1].minutos[index] := RTC_MINUTES;
        datos_evento[1].numero_eventos_hora := index;
end_if;


if R_TRIG_2.Q then
        index := datos_evento[2].numero_eventos_hora +1;
        datos_evento[2].minutos[index] := RTC_MINUTES;
        datos_evento[2].numero_eventos_hora := index;
end_if;


if R_TRIG_3.Q then
        index := datos_evento[3].numero_eventos_hora +1;
        datos_evento[3].minutos[index] := RTC_MINUTES;
        datos_evento[3].numero_eventos_hora  := index;
end_if;
```